

HANDBUCH



Reports für Intrex

Version 19.03

Ansprechpartner

QuinScape GmbH
Jasper Lauterbach
Wittekindstraße 30
D-44139 Dortmund
reports@quinscape.de
Tel.: +49 231 533 831 233
Fax.: +49 231 533 831 111

© 2019 QuinScape GmbH, Dortmund / Germany

quinscape.de

Alle Rechte vorbehalten

Ungeachtet der Sorgfalt, die auf die Erstellung des Textes, der Abbildungen und der Programme verwendet wurde, können weder der Verlag, der Autor, Herausgeber oder Übersetzer für mögliche Fehler und deren Folgen eine juristische Verantwortung oder irgendeine Art von Haftung übernehmen. Für Verbesserungsvorschläge und Hinweise ist der Herausgeber dankbar.

Das Werk ist in all Ihren Teilen urheberrechtlich geschützt. Alle Rechte vorbehalten, insbesondere das Recht der Übersetzung, des Vortrags, der Reproduktion, der Vervielfältigung auf fotomechanischen oder anderen Wegen und der Speicherung in elektronischen Medien.

Die Informationen in diesem Werk werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Alle wiedergegebenen Gebrauchs- und Handelsnamen, Warenbezeichnungen usw. können auch ohne besondere Kennzeichnung Marken sein und als solche den gesetzlichen Bestimmungen unterliegen.

Inhalt

1	Über Reports für Intrexx 19.03.....	8
1.1	Über dieses Dokument.....	8
1.2	Kompatibilität.....	8
2	Reports für Intrexx 19.03	9
2.1	Was ist neu in Version 19.03?.....	9
2.2	Key-Features.....	9
2.2.1	Einfache Integration	9
2.2.2	Geschwindigkeit	10
2.2.3	Parallele Verarbeitung.....	10
2.2.4	Direkte Auslieferung durch Servlets & Drill-Down	10
2.2.5	Globale Ressourcen, versioniert	10
2.2.6	Vielfältige Exportformate	11
2.2.7	Graphischer Berichtseditor	11
2.2.8	Zentrale Verwaltungsapplikation	11
2.2.9	Groovy-API.....	11
2.3	Parameter	12
2.3.1.1	Kontext-Objekte	12
2.3.1.2	Intrexx Formulardaten.....	15
2.3.1.3	Datengruppen-Daten	15
2.3.1.4	Freie Parameter im Prozess	15
2.4	Zeitonenbehandlung.....	15
2.5	Das Jaspersoft Studio & JasperReports	16
2.5.1	Das Jaspersoft Studio: Der neue Berichts-Designer	16
2.5.2	Aktualisierte JasperReports und Jaspersoft Studio Versionen.....	16
2.5.3	Schriftarten & formatierter Text	16
2.5.4	Report Book	19
2.6	Expert-Attribute für Schaltflächen-Integration.....	19
2.6.1.1	rq_qsReport.....	19
2.6.1.2	rq_qsShowInNewWindow	19
2.6.1.3	rq_qsShowOptions	20
2.6.1.4	rq_qsExportFormats	20
2.6.1.5	rq_qsExportFormat	20
2.6.1.6	rq_qsAutoshowReport	20
2.6.1.7	rq_qsTableRecords	20
2.6.1.8	rq_qsShowSql.....	20
2.6.1.9	rq_hasRun	20
2.6.1.10	rq_qsAsAttachment	20
2.6.1.11	rq_qsFileName	20
2.7	Die Groovy-API	21
2.7.1.1	context	22
2.7.1.2	debug.....	22
2.7.1.3	identifizier	22
2.7.1.4	datarecord.....	22
2.7.1.5	files	24
2.7.1.6	parameters.....	24

3	Neuinstallation und Update von vorheriger Version	25
3.1	Sicherung bestehender Daten.....	25
3.2	Wiederherstellung nach einem Portalimport.....	25
3.3	Probleme nach einem Update von Intrexx	26
3.4	Entfernen alter Bibliotheken, Reports für Intrexx 8.0.1	26
3.5	Entfernen alter Bibliotheken, Reports für Intrexx 8.0.0 Fehler! Textmarke nicht definiert.	
3.6	Entfernen alter Bibliotheken, Reports für Intrexx 7.0.3 Fehler! Textmarke nicht definiert.	
3.7	Entfernen alter Bibliotheken, Reports für Intrexx 7.0.1 Fehler! Textmarke nicht definiert.	
3.8	Entfernen alter Bibliotheken, Reports für Intrexx 6.0.4 Fehler! Textmarke nicht definiert.	
3.9	Entfernen alter Bibliotheken, Reports für Intrexx 6.0.3 Fehler! Textmarke nicht definiert.	
3.10	Entfernen alter Bibliotheken, Reports für Intrexx 6.0.2 Fehler! Textmarke nicht definiert.	
3.11	Entfernen alter Bibliotheken, Reports für Intrexx 6.0 Fehler! Textmarke nicht definiert.	
3.12	Update von einer älteren Version	27
3.13	Import des Anwendungspaket durchführen	27
3.14	Überprüfung der Konfigurationsdateien	28
3.15	Migrationsprozess ausführen.....	29
3.16	Reports für Intrexx – Beispieldaten und Prozess	29
4	Deinstallation	30
5	Ein erster Bericht	31
5.1	Ziel	31
5.2	Berichtserstellung mit dem Jaspersoft Studio	31
5.2.1	Installation und Konfiguration	31
5.3	Veröffentlichung des Berichts im Portal	44
5.4	Pakete.....	49
5.4.1	Export.....	50
5.4.2	Import.....	51
5.5	Aufruf des Berichts aus Ihrer eigenen Anwendung	52
5.5.1	Schaltflächen-Aufruf: Klassische Variante	52
5.5.1.1	Beispiel..... Fehler! Textmarke nicht definiert.	
5.5.2	Schaltflächen-Aufruf: Servlet.....	53
5.5.2.1	Import der JavaScript-Bibliothek	53
5.5.2.2	Beispiel	53
5.5.3	Prozessmanager-Aufruf	53
5.5.4	Groovy-API Aufruf	53
5.5.4.1	Beispiel 1	Fehler! Textmarke nicht definiert.
5.5.4.2	Beispiel 2	54
5.5.4.3	Beispiel 3	56
5.5.4.4	Beispiel 4	57
5.5.5	Drill-Down.....	58
6	Weiterführende Informationen.....	61
6.1	Jaspersoft University sowie Coaching und Unterstützung bei der Berichtserstellung	61
6.2	Managed Services: Support und verlängerte Werkbank für Ihre Berichtsanwendungen und Berichte.....	61
6.3	Jaspersoft Studio und JasperReports Dokumentation	62
6.4	Foren.....	62

6.5 Angrenzende Lösungen	63
6.5.1 Jaspersoft Business-Intelligence-Suite	63
6.5.2 Spotfire.....	64
A Anhang	65
A.1 Übersicht der verwendeten JasperReports-Version für Reports für Intrexx.....	65

Copyright

Das vorliegende Dokument ist in all seinen Teilen urheberrechtlich geschützt. Alle Rechte sind vorbehalten, insbesondere das Recht der Übersetzung, des Vortrags, der Reproduktion und der Vervielfältigung. Ungeachtet der Sorgfalt, die auf die Erstellung von Text, Abbildungen und Programmen verwendet wurde, können weder Autor, Herausgeber oder Übersetzer für mögliche Fehler und deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen.



Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. können auch ohne besondere Kennzeichnung Marken sein und als solche den gesetzlichen Bestimmungen unterliegen.

Schreibkonventionen

In diesem Dokument werden Textstellen *kursiv* dargestellt, wenn sie sich auf Einstellungen in den abgebildeten Dialogen beziehen. Menüpunkte, die in Kontextmenüs erreichbar sind, sind immer auch über das Hauptmenü erreichbar. Hauptmenüpunkte werden nicht beschrieben, es sei denn, sie sind nicht über das Kontextmenü erreichbar.

Programmiercode im Text wird in der Schriftart `Courier` dargestellt. Kontextmenüs können mit einem Klick mit der rechten Maustaste auf das beschriebene Element geöffnet werden.

`<intrex>` bezeichnet im Folgenden Ihren Intrex Installationspfad, unter Windows z.B.  `c:\intrex\`, unter Linux z.B.  `/opt/intrex/`.

Mit `<portal>` wird ein bestehendes Intrex Portalverzeichnis bezeichnet. Alle Dateien eines Portals finden Sie auf dem Intrex Portal Server in diesem Verzeichnis, unter Windows z.B.  `c:\ProgramData\intrex\<portal>`. Vor der Intrex Version 6 war dieses Verzeichnis unter  `<intrex>\org\` zu finden. Das Portalverzeichnis können Sie in den Portaleigenschaften, erreichbar im Intrex Portal Manager Hauptmenü *Portal*, ermitteln.

Folgende Symbole werden für die Kennzeichnung von speziellen Informationen verwendet:



Information



Verzeichnisse



URLs



Klick auf Schaltflächen

1 Über Reports für Intrexx 19.03

Reports für Intrexx 19.03 führt die Tradition der engen Integration der führenden Reporting Lösung JasperReports mit der Portalsoftware Intrexx fort. Ein ganz besonderes Augenmerk wurde dabei auf die Programmierbarkeit von Berichten im Prozessmanager gelegt.

Reports für Intrexx (vormals Reports für Intrexx) ermöglicht bereits seit dem Jahr 2004 die Erstellung professioneller Berichte und deren Verteilung auf Knopfdruck direkt aus Ihrem Intrexx-Portal. Dabei hat sich sowohl die Reporting-Engine JasperReports als auch die Integration mittels Reports für Intrexx stetig weiterentwickelt. Reports für Intrexx 19.03 stellt dabei einen weiteren Meilenstein in der Entwicklung dar.

Mit unserer langjährigen Erfahrung in der Berichtserstellung für Intrexx Portale haben wir, die QuinScape GmbH, Reports für Intrexx weiter überarbeitet und verfeinert. Auch die technologische Weiterentwicklung von Intrexx als Plattform ermöglicht es, neue Wege zu beschreiten.

Die aktuelle Version Reports für Intrexx 19.03 stellt dabei die Kompatibilität mit der aktuellen Intrexx Version 19.03 her und aktualisiert die Reporting-Komponenten auf den aktuellen Stand der Technik.

1.1 Über dieses Dokument

Dieses Dokument setzt einen sicheren administrativen Umgang mit der Portal-Software Intrexx 19.03 voraus. Das Niveau für notwendige Anpassungen bestehender Applikationen oder die Entwicklung neuer Applikationen setzt mindestens den Umfang der Schulung *Intrexx Advanced – Workflows & Processes* voraus. Der Entwurf von Berichten erfordert Kenntnisse in Datenbanken und deren Abfragen (SQL) sowie allgemeiner Datenstruktur-Konzepte von Intrexx.

1.2 Kompatibilität

Reports für Intrexx 19.03 setzt Intrexx 19.03 voraus. Bitte beachten Sie, dass für verschiedene Versionen von Intrexx auch passende Versionen von Reports für Intrexx existieren. Diese Reports für Intrexx-Versionen sind an die jeweilige Intrexx-Version gebunden und funktionieren nicht mit anderen Intrexx-Versionen. Eine detaillierte Übersicht der Versionen finden Sie im Anhang A.1.

2 Reports für Intrexx 19.03

Dieses Kapitel beleuchtet die Funktionen von Reports für Intrexx sowie die Möglichkeiten in der Berichtserstellung mittels des Jaspersoft Studios. Falls Sie noch keine Erfahrungen mit der Berichtserstellung mittels Reports für Intrexx haben, empfehlen wir zunächst die Lektüre des Kapitels 5 - Ein erster Bericht.

2.1 Was ist neu in Version 19.03?

Schulungen zur Benutzung des Jaspersoft Studios sowie für Reports für Intrexx im Speziellen können Sie im Rahmen der Jaspersoft University belegen. Weitere Informationen zur Jaspersoft University am Standort Dortmund finden Sie in Kapitel 6.1. Zudem können Sie zur Unterstützung bei der Berichtserstellung, zum Ausgleich von Ressourcenengpässen oder bei Problemstellungen auf unsere Supportdienstleistung zurückgreifen. Weitere Informationen zu unserer Supportdienstleistung finden Sie in Kapitel 6.2.

Neben der Kompatibilität zu Intrexx 19.03 wurde Reports für Intrexx auch **an die Version 6.9.0 von JasperReports angepasst**. Die neue Version ist nicht unbedingt abwärtskompatibel. Das heißt Berichte aus vorherigen Versionen laufen eventuell erst nach Anpassungen. Mehr Informationen dazu finden Sie in Kapitel 5.2. Eine Übersicht der verwendeten JasperReports Version für die jeweilige Reports für Intrexx Versionen finden Sie im Anhang A.1.

2.2 Key-Features

2.2.1 Einfache Integration

Reports für Intrexx ermöglicht die einfache Integration von Berichten in Ihre Applikationen. Dabei stehen Ihnen eine Reihe von Möglichkeiten der Integration zur Verfügung, viele davon ohne Programmierung:

1. Aufruf über Schaltflächen: Hierbei wird der zu generierende Bericht als Expert-Attribut der Schaltfläche definiert. Beispiele für die Integration mittels Schaltflächen finden Sie in den Kapiteln 5.5.1 und 5.5.2.
2. Aufruf im Prozessmanager: Der zu generierende Berichte kann wahlweise über Expert-Attribute einer Datengruppe und deren Datenfelder definiert werden oder mittels der neuen Groovy-API, die eine weitaus flexiblere Integration ermöglicht Beispiele für die Integration im Prozessmanager finden Sie in den Kapiteln 2.6 und 5.5.3.
3. HTTP-API: Über ein spezielles Intrexx Servlet steht Ihnen auch die Erzeugung und Parametrisierung eines Berichts mittels URL Aufruf zur Verfügung.

2.2.2 Geschwindigkeit

Ein Hauptaugenmerk bei der Erstellung von Berichten liegt immer auf der Geschwindigkeit, mit der der Bericht erzeugt und ausgeliefert wird. Reports für Intrexx und die zugrunde liegende Reporting Engine JasperReports erledigt dies in der Regel in für den Benutzer kaum wahrnehmbarer Zeit. Standard-Druckansichten werden i.d.R. in weniger als einer Sekunde erzeugt. Selbst komplexere Berichte verursachen i.d.R. keine langen Wartezeiten.

2.2.3 Parallele Verarbeitung

In größeren Installationen tritt vermehrt die Situation auf, dass viele Benutzer gleichzeitig ein und denselben Bericht aufrufen, bspw. wenn ein Lieferschein oder ein Anschreiben als Bericht implementiert wurde. Anwender müssen aber nicht aufeinander warten, sondern können ihre Berichte parallel vom System erzeugen lassen.

2.2.4 Direkte Auslieferung durch Servlets & Drill-Down

Häufig werden Berichte aufgerufen, um schnell und einfach gezielt Informationen aus einem Portal auszudrucken. Hierzu muss das generierte Dokument nicht archiviert oder auf dem Server abgespeichert werden. Über die Einbindung eines Berichts als Schaltfläche in Ihrer Applikation können Sie Berichte im „streaming“-Verfahren als Datenströme direkt an den Browser eines Benutzers ausliefern. Wie Sie einen Bericht so ausliefern können, erfahren Sie im Kapitel 5.5.2 auf Seite 8.

Die Auslieferung eines Berichts durch ein Servlet hat einen weiteren entscheidenden Vorteil: Es kann ein echtes Drill-Down erfolgen, in dem im Bericht via Hyperlinks die nächste Informationsebene parametrisiert über die HTTP-API des Servlets angesteuert werden kann.

2.2.5 Globale Ressourcen, versioniert

Um ein zentrales Corporate Design durchzusetzen muss es einfach sein, zentral Logos oder Kopf- und Fußzeilen in Ihren Berichten auszutauschen. Dies ist über die zentrale Ressourcenhaltung in Reports für Intrexx möglich. Definieren Sie einmalig Ihre Corporate Design Elemente und verwenden Sie diese in Ihren Berichten als eingebundene Subreports oder Bilder immer wieder. Wenn Sie dann den zentrale Fußzeilenbericht oder das Logo austauschen gilt dies automatisch für alle Berichte, die diese verwenden. Bilder müssen nur einmal publiziert werden und können in verschiedenen Berichten wiederverwendet werden.

Ressourcen werden bei jeder Änderung gesichert; Sie können mit einem Mausklick eine vorige Version wiederherstellen.

2.2.6 Vielfältige Exportformate

Neben dem Standardformat „PDF“ unterstützt Reports für Intrexx auch die aktuellen Office Formate `docx`, `xlsx`, `pptx` sowie die OpenOffice Formate `odf/odt` und `ods`. Insgesamt stehen damit auch offene, nicht proprietäre Formate für die Textverarbeitung und Tabellenkalkulation sowie gänzlich neue Möglichkeiten im Bereich von Präsentationen zur Verfügung.

Es werden derzeit die folgenden Exportformate angeboten:

- | | | |
|---------|---------|------------|
| 1. PDF | 5. DOCX | 9. ODF/ODT |
| 2. XLS | 6. PPTX | 10. ODS |
| 3. XLSX | 7. HTML | |
| 4. CSV | 8. RTF | |

2.2.7 Graphischer Berichtseditor

Berichte können, ähnlich wie Intrexx Applikationen, über einen graphischen Designer – Jaspersoft Studio – gestaltet werden. Mittels Assistenten können die vielfältigen Report-Elemente einfach im Bericht verwendet und durch mächtige Ausrichtungswerkzeuge schnell positioniert werden.

2.2.8 Zentrale Verwaltungsapplikation

Um in Intrexx die Berichte mit ihren zugehörigen Ressourcen (z.B. Logos, Berichtsdefinitionen) „bekannt“ zu machen, werden diese in einer eigenen Intrexx Applikation verwaltet. Sie können einzelne Ressourcen in der Applikation hochladen und dann zu Berichten zusammenfassen. Auch können Sie in den meisten Fällen die Berichte direkt in der Applikation für die verschiedenen Ausgabeformate testen. Dabei werden die im Bericht verwendeten Parameter automatisch ausgelesen und abgefragt. Wenn der Parameter in einem Eingabefeld eingegeben werden kann, kann dieser für den Test verwendet werden.

Der Aufbau und die Handhabung der Applikation werden im Kapitel 5.3 ab Seite 44 erläutert.


2.2.9 Groovy-API

Die Groovy-API gibt volle Kontrolle über die Berichtsgenerierung im Prozessdesigner. Bestimmen Sie mittels Groovy, welcher Bericht erstellt wird, wie die Ausgabedatei heißen soll, in welcher Applikation, Datengruppe und Datenfeld der Bericht in welchem Format gespeichert werden soll und erstellen Sie komplexe, berechnete Parameter. Eine vollständige Beschreibung der API finden Sie im Kapitel 2.7.


2.3 Parameter

Fast alle Berichte müssen parametrisiert werden, um sinnvolle Ausgaben zu ermöglichen. Dies kann ein bestimmter Datensatz oder eine Menge von Filterbedingungen sein. Es können in Berichten vielfältige Parameter genutzt werden.





- Kontext-Objekte
- Intrexx Formulardaten
- Datengruppen-Daten
- Komplexe, programmierte Objekte

 **Hinweis:** Alle Parameternamen sind **case-sensitive**. Das bedeutet, dass in Bezug auf Groß- und Kleinschreibung die Schreibweise exakt übereinstimmen muss.

2.3.1.1 Kontext-Objekte

 **Hinweis:** Folgende Parameternamen bzw. -präfixe sind durch die Verwendung der impliziten Parameter reserviert und dürfen nicht in Ihren Berichten verwendet werden:

- CONNECTION_
- REPORT_
- INTREXX_
- JASPER_

 **Hinweis:** Um die folgenden impliziten Parameter [`de.uplanet.<Parameterdetails>`] im Jaspersoft Studio verwenden zu können muss die Intrexx Bibliothek `ix-server.jar` aus dem Verzeichnis  `<intrexx>\lib` im Klassenpfad des Jaspersoft Studios zur Verfügung stehen. Falls Sie das Jaspersoft Studio nicht auf dem Server betreiben wird empfohlen, die `.jar`-Datei `ix-server.jar` aus dem Verzeichnis  `<intrexx>\lib` auf Ihren lokalen Rechner in ein eigenes Verzeichnis zu kopieren und den Klassenpfad um diese Bibliothek zu erweitern. Über einen *Rechtsklick* auf Ihr Jaspersoft Studio Projekt und Auswahl der *Properties*, kann diese Bibliothek unter *Java Build Path, Libraries*,  *Add JARs* hinzugefügt werden.

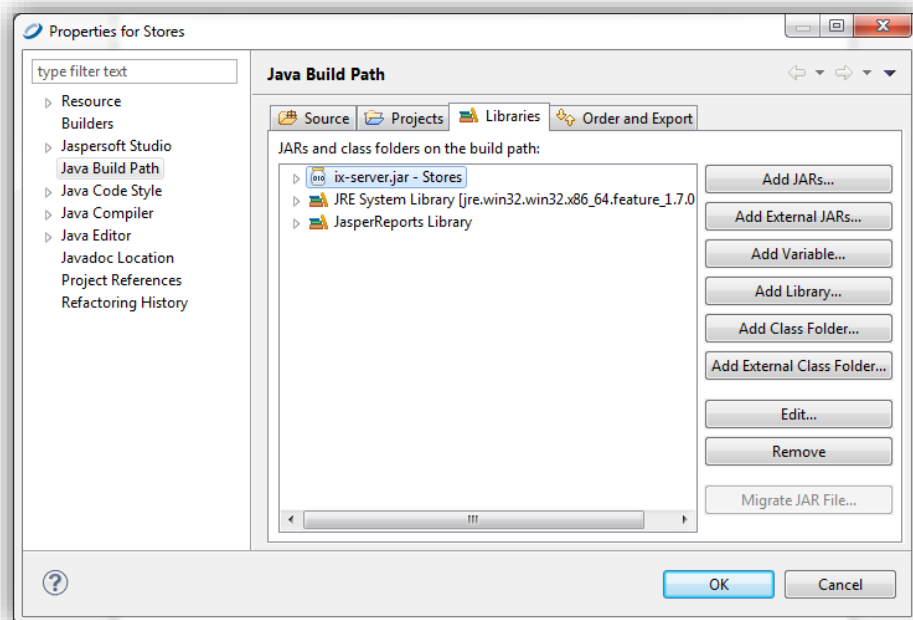


Abbildung 1 ix-server.jar zum Java Build Path hinzufügen

2.3.1.1.1 INTREXX_SESSION

Wird ein Parameter mit diesem Namen im Bericht definiert, wird die Intrexx-Session als

```
de.uplanet.lucy.server.session.ISession
```

übergeben, sofern vorhanden. Die Intrexx Session enthält Informationen zur aktuellen Benutzersitzung, wie bspw. die Session ID oder kontextabhängige Informationen.

2.3.1.1.2 REPORT_CONTEXT

Für einen Parameter mit diesem Namen wird der komplette Kontext des aktuellen Berichts als

```
de.quinscape.intrexx.report.ReportContext
```

übergeben.

2.3.1.1.3 INTREXX_CONTEXT

Für einen Parameter mit diesem Namen wird der komplette Intrexx-Kontext des aktuellen Berichts als

```
de.uplanet.lucy.server.businesslogic.IBusinessLogicProcessingContext
```


übergeben.

2.3.1.1.4 INTREXX_USER

Für einen Parameter mit diesem Namen wird der aktuelle Intrexx-Benutzer als

```
de.uplanet.lucy.server.usermanager.IUser
```

übergeben, sofern vorhanden.

 **Achtung:** Bei der Verwendung im Rahmen des Prozessmanagers ist nicht unbedingt genau definiert, welcher Benutzer hier übergeben wird. (z.B. beim Aufruf via globalen Timern).

2.3.1.1.5 INTREXX_REQUEST

Für einen Parameter mit diesem Namen wird der aktuelle Intrexx-Request als

```
de.uplanet.lucy.server.connector.IServerBridgeRequest
```

übergeben, sofern vorhanden. Aus dem Request kann bspw. die REQUEST_URL ausgelesen werden, um Hyperlinks für Drill-Down-Funktionen in Berichten zu ermöglichen.

2.3.1.1.6 INTREXX_SHARED_STATE

Für einen Parameter mit diesem Namen wird der aktuelle

```
de.uplanet.lucy.server.SharedState
```

übergeben. Der Shared-State ist ein neues Kontext-Objekt, in dem während einer Request-Abarbeitung oder einer Prozess-Ausführung Daten hinterlegt werden können.

2.3.1.1.7 INTREXX_LANGUAGE

Für einen Parameter mit diesem Namen wird der zweistellige ISO-Code der aktuellen Sprache als

```
java.lang.String
```

übergeben. Die aktuelle Sprache wird entweder aus dem Request (vom Benutzer aktuell eingestellt) oder anhand des Benutzers (Einstellung des Benutzers in der Benutzerverwaltung) ermittelt. Im Zweifelsfall wird die Standard-Sprache des Portals geliefert.

2.3.1.1.8 INTREXX_USER_TIME_ZONE

Für einen Parameter mit diesem Namen wird die für den Benutzer eingestellte Zeitzone als

```
java.util.TimeZone
```

übergeben. Liefert im Zweifelsfall die Standardzeitzone des Portals.

2.3.1.1.9 CONNECTION_*

Für Parameter, die mit diesem Namen beginnen, wird versucht eine Datenbankverbindung aus dem Kontext zu übergeben. Gesucht wird die Datenbankverbindung mit dem Parameternamen (ohne das Präfix). Für den Parameter mit dem Namen `CONNECTION_IxSysDb` wird also die Verbindung zur Systemdatenbank des Portals übergeben. Sämtliche Datenbankverbindungen werden im Intrexx Portal Manager im Bereich *Integration, Datenquellen* definiert. Wurde dort bspw. eine Verbindung zu Ihrem ERP System mit dem Namen `ERP` definiert, können Sie mittels eines Parameters `CONNECTION_ERP` (vom Typ `java.sql.Connection`) im Bericht auf diese Verbindung zugreifen. Die bislang erforderliche „Schattenverwaltung“ ist nun nicht länger erforderlich. Insbesondere können somit verschiedene Datenbankverbindungen in einem Bericht verwendet werden.

2.3.1.2 Intrexx Formulardaten

Wenn Sie Daten aus einem Formular an einen Bericht übergeben möchten, können Sie dies mit den Standard-Eingabefeldern von Intrexx bewerkstelligen. Dazu müssen Sie lediglich das NAME-Attribut des Eingabefelds so benennen, wie der Parameter im Bericht benannt ist. Dadurch wird die Name des Objekts im Request entsprechend benannt und Reports für Intrexx kann eine Verbindung zwischen der Eingabe des Benutzers und dem erwarteten Parameter herstellen. Ein Beispiel finden Sie in Kapitel 5.5.1 und 5.5.2.

2.3.1.3 Datengruppen-Daten

Sollen Daten serverseitig ermittelt und für die Parametrisierung von Berichten verwendet werden, können diese über Sysidents an die Berichts-Parameter gekoppelt werden. Ein Beispiel finden Sie in Kapitel 5.5.3.

2.3.1.4 Freie Parameter im Prozess

Über die neue Groovy-API können Sie selbst bestimmen, welche Parameter Sie mit welchen Inhalten füllen möchten. Dies können bspw. berechnete oder zusammengesetzte Daten sein. Ein Beispiel finden Sie in Kapitel 0.

2.4 Zeitonenbehandlung

Intrexx speichert Datumswerte in der Zeitzone „UTC“ ab. Um diese korrekt im Report auszugeben mussten bislang mittels Datumsberechnungen das richtige lokale Datum ermittelt werden. Nun steht hierfür eine Property innerhalb von JasperReports zur Verfügung, mit der die Zeitzone wahlweise für den gesamten Bericht, für ein Dataset oder für ein einzelnes Datenfeld oder einen einzelnen Parameter gesetzt werden kann.

Vergeben Sie dazu auf dem entsprechenden Objekt den Parameter

```
net.sf.jasperreports.jdbc.time.zone
```

und verwenden als Wert die gewünschte Zeitzone (für Intrexx ist dies in der Regel „UTC“ (ohne die Anführungszeichen).

Sie können zudem einen Parameter mit dem Namen:

```
net.sf.jasperreports.jdbc.time.zone
```

und einem entsprechenden Wert an den Report übergeben. Dies übersteuert dann alle entsprechenden Property-Einstellungen im Report.

2.5 Das Jaspersoft Studio & JasperReports

Nicht nur Reports für Intrexx entwickelt sich kontinuierlich weiter. Auch die Reporting-Engine JasperReports und der dazu gehörige grafische Editor das Jaspersoft Studio unterliegen einem kontinuierlichen Verbesserungsprozess. Im Folgenden werden einige neue Funktionen hervorgehoben. Weiterführend Informationsquellen finden Sie im Kapitel 6.3 ab Seite 62.

2.5.1 Das Jaspersoft Studio: Der neue Berichts-Designer

Seit der JasperReports Version 5.5.0 ist das Jaspersoft Studio der offizielle Berichtsdesigner für JasperReports. Der iReport Designer wurde nur noch bis Ende 2015 offiziell durch den Hersteller TIBCO Jaspersoft unterstützt und kritische Fehler wurden bis dahin behoben. Daher haben wir bereits mit Reports für Intrexx 7 den Berichtsdesigner vom iReport Designer auf das Jaspersoft Studio umgestellt, damit Sie weiterhin von allen Neuerungen profitieren können und Ihre Software nicht veraltet.

Das Jaspersoft Studio ist ein auf Eclipse basierender Berichtsdesigner für JasperReports und den JasperReports Server und eine komplette Neufassung des iReport Designers. Es ist sowohl als Eclipse Plug-In als auch als Stand-Alone-Anwendung verfügbar. Personen die bereits mit Eclipse gearbeitet haben, sollten mit dem Benutzerinterface schnell zu Recht kommen.

2.5.2 Aktualisierte JasperReports und Jaspersoft Studio Versionen

Mit Reports für Intrexx 19.03 werden die Version 6.9.0 von JasperReports und dem Jaspersoft Studio unterstützt. Da durch diese Umstellung ein recht großer Versionsprung durchgeführt wurde, stehen eine Reihe von neuen Funktionalitäten zur Verfügung. Eine vollständige Auflistung kann in dem jeweiligen Change Log (Readme) unter <http://community.jaspersoft.com/project/jasperreports-library/releases> eingesehen werden.

2.5.3 Schriftarten & formatierter Text

Innerhalb des Jaspersoft Studios können eigene Schriftarten-Pakete auf Basis von True-Type, Embedded Open Type, Scalable Vector Graphics oder Web Open Font Format

Schriften zusammengestellt werden, die von der Reporting-Engine korrekt im Hinblick auf Textformatierung (bspw. Fettdruck) in der PDF-Ausgabe behandelt wird.

Hinweise: Wenn Sie Schriftarten zum Jaspersoft Studio hinzufügen, müssen Sie diese in jede Ihrer Jaspersoft Studio Installationen hinzufügen.

Um eine Schriftarten-Bibliothek für Ihre Berichte zu definieren, gehen Sie bitte wie folgt vor. Lassen Sie sich nicht von dem Umfang dieses Beispiels abschrecken. In der Regel muss dies nur einmalig für die zu verwendenden Schriftarten durchgeführt werden. Aus lizenzrechtlichen Gründen ist es leider nicht möglich, bereits vorkonfigurierte Schriftarten-Bibliotheken, außer dem vorliegenden Beispiel, beizufügen. Die Schriftarten können für den gesamten Workspace konfiguriert werden oder für jedes Projekt separat.

1. Öffnen Sie den Editor – das Jaspersoft Studio – und wechseln Sie in die Optionen des Projektes (🖱️ *Project, Properties*).
2. Wechseln Sie in den Bereich 🖱️ *Fonts* (unterhalb von *Jaspersoft Studio*).

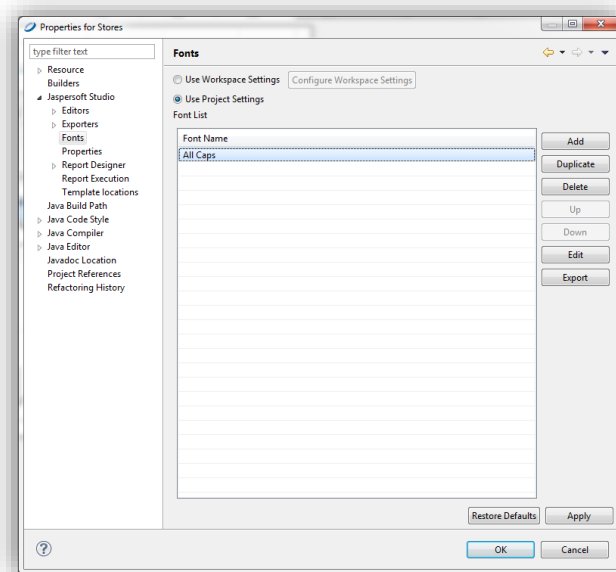



Abbildung 2 Font-Einstellung im Jaspersoft Studio

3. Das Jaspersoft Studio bietet hier einen Assistenten, mit dem einerseits TrueType Schriftarten (Alternativ auch Embedded Open Type, Scalable Vector Graphics oder Web Open Font Format Schriftarten) für die Verwendung in Ihren Berichten bereitgestellt werden können und über den andererseits auch eine entsprechende Bibliothek (eine sog. Extension) für den Einsatz in Ihrem Intrexx Portal

erzeugt werden kann. Klicken Sie für den Assistenten auf die Schaltfläche  **Add**.

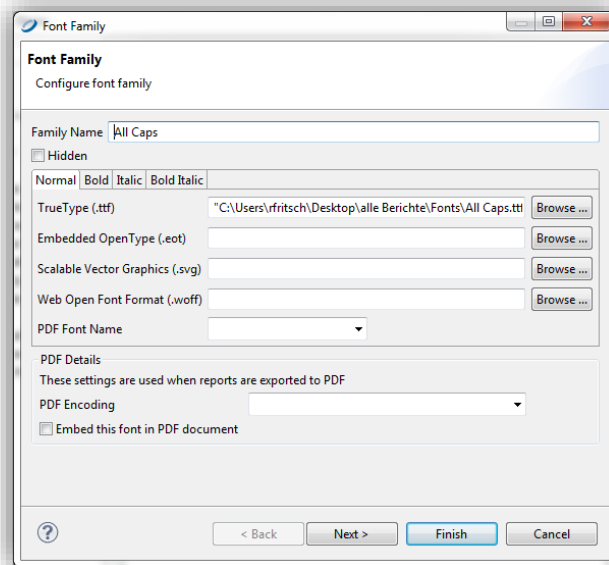




Abbildung 3 Font-Assistent im JasperSoft Studio Schritt 1

4. Mittels einer der  **Browse** Schaltflächen können Sie die Haupt-Schriftarten-Datei auswählen (der normale Zeichensatz).
5. Im Folgenden wählen Sie dann die passenden Schriftarten-Dateien für Fett-druck, Kursiv und Kursiv-Fett in den einzelnen Reitern (Bold, Italic, Bold Italic).
6. In den nächsten Schritten können Sie noch Ersetzungen (Font Mapping) definieren und die Verwendung auf bestimmte (Locales) Regionen einschränken. Dies ist in der Regel nicht nötig.
7. Abschließend ist die Schrift-Familie definiert und kann in den Schriftarten-Dialogen im entsprechenden Projekt im JasperSoft Studio verwendet werden. Damit diese Schrift auch in IntrexX bereitgestellt werden kann, muss sie noch exportiert werden.
8. Klicken Sie auf die Schaltfläche  **Export**. Dies öffnet einen Datei-Dialog, in dem Sie den Namen und den Speicherort angeben können.

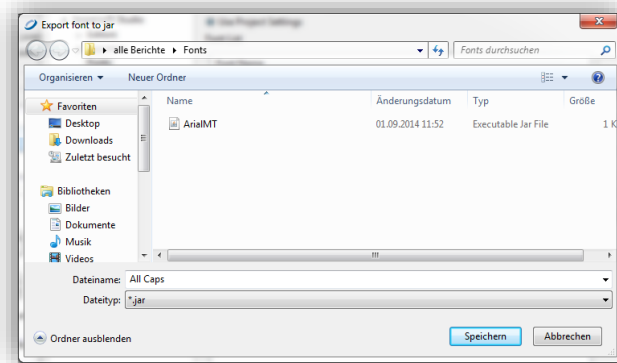



Abbildung 4 Benennung des Extension-Exports

9. Kopieren Sie die angelegte .jar Datei in Ihr Portal-Lib Verzeichnis ( <portal>/lib) und starten Sie Ihr Portal neu. Die Schriftart steht nun für Reports im Portal zur Verfügung.

2.5.4 Report Book

Mit dem Report Book können mehrere Reports oder Subreports in einem Bericht gebündelt werden. Ein „Report Book“ wird als einzelne .jrxml Datei generiert und nutzt nicht die bekannte Aufteilung in Bänder, sondern arbeitet mit Abschnitten. In diesen Abschnitten können Reports, Subreports, Titelseiten oder Inhaltsverzeichnisse organisiert werden. Über das „Report Book“ ist daher bspw. die Erzeugung von Dokumenten mit variablen Seitengröße oder Ausrichtungen möglich.

Weitere Details können im unter folgender URL eingesehen werden:

<http://community.jaspersoft.com/documentation/tibco-jaspersoft-studio-user-guide/v60/report-books>

2.6 Parameter für die Report-Erzeugung

Die folgenden Attribute werden unterstützt (zur Übergabe siehe bitte Kapitel 5.5.2 *JavaScript-Aufruf: Servlet*):

2.6.1.1 rq_qsReport

Dieses Attribut muss zwingend angegeben werden. Es steuert, welcher Bericht erzeugt wird. Es kann wahlweise der Berichtsname oder dessen GUID angegeben werden.

2.6.1.2 rq_qsShowInNewWindow

Wird nicht mehr unterstützt, da Intrexx 19.03 keine Popup-Fenster mehr erlaubt.

2.6.1.3 rq_qsShowOptions

Wird nicht mehr unterstützt, da Intrexx 19.03 keine Popup-Fenster mehr erlaubt.

2.6.1.4 rq_qsExportFormats

Wird nicht mehr unterstützt, da Intrexx 19.03 keine Popup-Fenster mehr erlaubt.

2.6.1.5 rq_qsExportFormat

Steuert, in welchem Ausgabeformat der Report erzeugt werden soll. Unterstützt werden:

- | | | |
|---------|---------|---------|
| 1. PDF | 5. DOCX | 9. ODT |
| 2. XLS | 6. PPTX | 10. ODS |
| 3. XLSX | 7. HTML | |
| 4. CSV | 8. RTF | |

2.6.1.6 rq_qsAutoshowReport

Wird nicht mehr unterstützt, da Intrexx 19.03 keine Popup-Fenster mehr erlaubt.


2.6.1.7 rq_qsTableRecords

Wird nicht mehr unterstützt, da Intrexx 19.03 keine Popup-Fenster mehr erlaubt.

2.6.1.8 rq_qsShowSql

Wird nicht mehr unterstützt, da Intrexx 19.03 keine Popup-Fenster mehr erlaubt.

2.6.1.9 rq_hasRun


 **Hinweis:** Das Attribut `rq_hasRun` wird nicht mehr benötigt. Aus Kompatibilitätsgründen wird dieser Attributname noch unterstützt, sollte aber nicht mehr verwendet werden.

2.6.1.10 rq_qsAsAttachment

Wird nicht mehr unterstützt, da Intrexx 19.03 keine Popup-Fenster mehr erlaubt.

2.6.1.11 rq_qsFileName

Dieses Attribut steuert bei einem Schaltflächen-Servlet-Aufruf (siehe 5.5.2) den Dateinamen, der beim Download bzw. im Dialog „Datei speichern unter ...“ vorgegeben wird.


 **Hinweis:** Dieses Expert-Attribut steht nur in Verbindung mit dem JavaScript-Servlet-Aufruf (siehe Kapitel 5.5.2) zur Verfügung.

2.7 Die Groovy-API

Die Groovy-API ist die flexibelste Möglichkeit innerhalb eines Prozesses einen Bericht zu erzeugen, da die Groovy-API für den Aufruf eines Berichts eine Vielzahl von Parametern bereitstellt. Als prägnantes Beispiel ist es mit ihr nun beispielsweise mit nur einer Angabe möglich, den Dateinamen eines Berichts vorzugeben – was vorher nur mit deutlich mehr Aufwand möglich war. Außerdem ist es möglich geworden, vor dem Aufruf des Berichts weitere Daten innerhalb des Prozesses zu ermitteln und diese direkt in den Report hinzureichen, was vorher nur durch Umwege möglich war.


Den konkreten Aufruf der API innerhalb eines Prozesses können Sie der Groovy-Bibliothek oder dem Beispiel aus Kapitel 0 entnehmen; an dieser Stelle werden nur die möglichen Parameter genannt und erklärt.

Die Technik hinter der Groovy-API stützt sich auf den angegebenen Kontext (vgl. Kapitel 2.7.1.1) und bezieht – soweit möglich – die entsprechenden Parameter aus diesem Kontext. Enthält der Kontext beispielsweise einen Datensatz, da die Groovy-Aktion im Rahmen eines Datengruppen-Ereignisbehandlers verwendet wird, so wird die Datengruppe und der Datensatz daraus automatisch ermittelt.

 **Erläuterung von Groovy Maps:** Der Aufruf selbst basiert auf sogenannten Maps, die in Groovy durch eckige Klammern eingeleitet und geschlossen werden und eine kommagetrennte Liste von Wertepaaren enthält. Hier ein Beispiel einer solchen verschachtelten Liste:

```
[
  apfel: "gruen",
  zwei: 2,
  wahrheit: true,
  mapInMap: [
    banane: "gelb"
  ],
  listeInMap: [1, 2, 3],
  datum: new Date()
]
```

Zeilenumbrüche und Einrückungen sind an diesen Stellen nicht notwendig, aber der Lesbarkeit halber zu bevorzugen. Ein Wertepaar besteht aus einem Parameternamen, einem Doppelpunkt sowie einem zugeordneten Wert eines bestimmten Java-Datentyps. Diese Datentypen sind damit analog zu denen im Jaspersoft Studio aufgrund der Gemeinsamkeit „Java“.

 **Schreibweisen:** Alle Parameternamen werden in Groß- und Kleinschreibung unterschieden, sodass ein genauer Blick auf die Namen gelegt werden sollte. Alternativ lassen sich auch aus der Groovy-Bibliothek passende Abschnitte einfügen.

2.7.1.1 context

Dies ist der wichtigste und einzige nicht optionale Parameter: Hier wird der entsprechende Intrexx-Portal-Kontext überreicht – dieser Parameter muss aber eigentlich nie angepasst werden, sodass Sie das `g_context` aus den Beispiel-Aufrufen einfach übernehmen können.

Datentyp: `IBusinessLogicProcessingContext`

2.7.1.2 debug

Dieser Parameter führt bei Aktivierung dazu, dass alle gesammelten Daten (aufzurufender Report, eingereichte Parameter, ...) in der Logdatei `portal.log` protokolliert werden. Voraussetzung dafür sind die Anpassungen aus dem Kapitel 3.7 (Überprüfung der Konfigurationsdateien). Diese Logdatei befindet sich im Portalverzeichnis im Verzeichnis `log` und wird direkt beim Aufruf befüllt. Etwaige Konfigurationsfehler lassen sich so wesentlich leichter nachvollziehen.

Datentyp: `Boolean`

2.7.1.3 identifier

Hiermit wird der aufzurufende Bericht definiert. Möglich ist dies zum einen durch die GUID des Berichts und zum anderen auch über den angegebenen Namen innerhalb der Reportsapplikation.

Datentyp: `String`

2.7.1.4 datarecord

Dies sind die Speicherorte des Berichts – also in diesem Fall Datensätze in einer Datengruppe. Für diese Variante gibt es vielerlei Einstellungsmöglichkeiten; zum Beispiel sind unterschiedliche Formate in verschiedenen Dateifeldern realisierbar.

Datentyp: `Map` (Hier beginnt also eine Verschachtelung.)

2.7.1.4.1 newRecord

Dieser Parameter gibt an, ob ein neuer Datensatz erzeugt werden soll. Falls es sich um eine Datengruppe mit zusammengesetztem Primärschlüssel handelt, so muss für diesen neuen Datensatz in dieser Datengruppe der Parameter `recordId` auch angegeben werden, da Intrexx diese Primärschlüssel nicht automatisch generiert.

Datentyp: `Boolean`

2.7.1.4.2 recordId

Hiermit wird angegeben an welchen Datensatz die Reports angefügt werden sollen, Für einen neuen Datensatz ist daher der neue Parameter `newRecord` zu bevorzugen, wobei aus historischen Gründen auch der Wert `-1` für einen neuen Datensatz möglich ist.



Zusammengesetzte Primärschlüssel: Bei zusammengesetzten Primärschlüsseln ist dieser Parameter erforderlich und wird mit der üblichen Intrexx-Konvention (zum Beispiel `LANG= de, LID= 1`) angegeben. An dieser Stelle sind technisch bedingt zwingend die Datenbankspalten-Bezeichner zu verwenden.

Datentyp: String (Zahlen werden an dieser Stelle automatisch zu einem String umgewandelt.)

2.7.1.4.3 recordValues

Hier können Sie weitere Werte des **neuen** Datensatzes angeben, sodass dieser außer dem Report und den Systemfeldern (einfügender Benutzer, Erstellungsdatum, ...) weitere Werte direkt nach der Erstellung enthält. So ist es zum Beispiel auch möglich den neuen Report unter einen vorhandenen Datensatz einzuhängen, indem man die Elternfelder (in der Regel `FKLID`) angibt. Der Aufbau besteht aus einer Map mit dem Spaltennamen (oder dem Sysident oder der Guid des jeweiligen Felds) und dem zuzuweisenden Wert:

```
[FKLID: 3, STR_TITEL: "Test"]
```

Datentyp: Map

2.7.1.4.4 datagroupGuid

Dieser Parameter bestimmt die Datengruppe, in der die Reports hinterlegt werden sollen. Dies ermöglicht es zum Beispiel Reports mit Parametern aus einer Datengruppe A aufzurufen, den Bericht aber in einer Datengruppe B zu hinterlegen – wobei man dann auch sinnvollerweise den Parameter `recordId` verwenden sollte, d.h. die datensatz-ID angeben sollte, unter der der Bericht abgelegt wird, damit es nicht zu Missverständnissen oder gar Fehlern kommt.

Datentyp: String

2.7.1.4.5 datafields

Dies spezifiziert die Dateifelder, in die letztendlich der Report hinterlegt wird – wobei für jedes Format nur ein Dateifeld pro Groovy-API-Aufruf möglich ist. Jeder Eintrag der Collection besteht wiederum aus einer Map, die im Wesentlichen wie folgt aussieht:

```
[format: "PDF", filename: "Test", identifier: "FILE_DATENFELD_001B5D85"]
```

- Hierbei sind als Identifier die GUID, der Spaltenname oder der Sysident des Zielfelds erlaubt.
- Die Formate entsprechen denen aus Kapitel 2.6.1.4 (`rq_qsExportFormats`).

- Alle drei Angaben sind vom Typ String.
- Der Dateiname ist optional – falls er nicht angegeben wird, wird der entsprechende Berichtsname verwendet. Es sind also mindestens `format` und `identifier` anzugeben, damit eine eindeutige Zuordnung möglich ist.
- Für jedes Exportziel bzw. jedes Dateifeld in das ein Bericht abgelegt werden soll muss eine solche Map definiert werden. Wenn es mehrere dieser Ziele gibt, so werden diese Maps durch Kommata getrennt.

Datentyp: Collection (Aufbau vgl. vorstehender Absatz)

2.7.1.5 files

Mit Hilfe dieses Parameters ist es möglich, dass generierte Reports direkt auf dem Dateisystem gespeichert werden. Jeder Eintrag der Collection besteht wiederum aus einer Map, die wie folgt aussieht:

```
[format: "PDF", path: "D:/", filename: "Test"]
```

- Die möglichen Formate entsprechen den in Kapitel 2.6.1.4 aufgelisteten Formaten.
- Die drei Parameter `format`, `path` sowie `filename` sind vom Typ String.
- Der Dateiname ist eine optionale Angabe; sofern dieser nicht angegeben wird, wird der entsprechende Berichtsname verwendet.
- Die Parameter `format` sowie `path` sind verpflichtende Angaben.
- Für jede abzulegende Datei muss eine Map in beschriebener Form definiert werden. Sollen mehrere Zieldateien abgelegt werden, so werden diese Maps durch Kommata getrennt.

Datentyp: Collection (Aufbau vgl. vorstehender Absatz)

2.7.1.6 parameters

Hiermit werden die Parameter für den Report bestimmt – dabei gibt es mehrere Varianten für die Belegung von Parametern. Die einfachste und gebräuchlichste Variante sind Werte aus dem Datensatz via `g_record["..."]`. Der Zusatz `getValue()` – also `g_record["..."].getValue()` – ist nicht nötig, da der Value-Holder vor dem Hineinreichen „entpackt“ wird. Alternativ können natürlich auch Zahlen, Strings oder andere Werte aus der Intrexsumgebung (zum Beispiel Requestwerte via `g_request["..."]`) verwendet werden. Die exemplarische Map aus der Einleitung dieses Kapitels ist beispielsweise eine gültige Parameter-Map, die sechs verschiedene Parameter in den Report hineinreicht.

Datentyp: Map

3 Neuinstallation und Update von vorheriger Version

Eine Neuinstallation unterscheidet sich nur unwesentlich von dem Update einer vorherigen Version. Bei einer Neuinstallation sind die lediglich die Kapitel 3.1 und 3.6 bis 3.7 relevant. Bei einem Update muss das Portal vorher gestoppt werden, damit das Löschen der jeweiligen Dateien möglich ist. Führen Sie ein Update über mehrere Versionen durch, so sind die Kapitel 3.4 bis **Fehler! Verweisquelle konnte nicht gefunden werden.** je nach Ausgangsversion kumulativ zu berücksichtigen.



Hinweis: Abweichend zur bisherigen Installationsweise ist kein separater Installer mehr notwendig.

In Kapitel 3.9 wird eine optional zu installierende Applikation mit Beispielen für die verschiedenen Integrationsmöglichkeiten von Berichten aus Ihren Anwendungen/Ihrem Portal heraus vorgestellt. Sofern Sie ein Update durchführen und die Beispielapplikation weiterhin verwenden möchten, so müssen Sie auch diese einmal neu importieren.



Hinweis: Falls Sie bislang mit der Property-Anpassung

`QS_TIME_ZONE_ID`

gearbeitet haben, müssen Sie diese in Ihren Berichten gegen die nun offizielle Zeitzonebehandlung austauschen. Die von QuinScape entwickelte Lösung für die Zeitzoneproblematik wurde seitens Jaspersoft in den offiziellen Standard überführt, jedoch mit einem anderen Bezeichner versehen. Um Software-Konflikten vorzubeugen, musste der `QS_TIME_ZONE_ID`-Fix aus Reports für Intrexx weichen. Das aktuelle Vorgehen können Sie in Kapitel 2.4 nachlesen.

3.1 Sicherung bestehender Daten


Wie immer sollte auch bei dieser Installation eine Sicherung des bestehenden Portals durchgeführt werden.



Achtung: Sofern Sie eigene Businesslogik, Treiber o.ä. (.jar Dateien) in diesem Verzeichnis abgelegt haben, überprüfen Sie bitte, ob die zu löschenden Bibliotheken im Konflikt mit von Ihnen verwendeten Bibliotheken stehen. In diesem Fall empfehlen wir eine Kontaktaufnahme mit uns unter reports@quinscape.de.

3.2 Wiederherstellung nach einem Portalimport

Sofern Sie ein Portal mit Reports für Intrexx importieren, müssen die durch Sie vorgenommenen Anpassungen an Konfigurationsdateien erneut durchgeführt werden.

Überprüfen Sie dazu die Konfigurationsdateien `portal.wcf` und `log4j.properties` im Verzeichnis  `<intrexx>\org\<portal>\internal\cfg`.

Die Datei *portal.wcf* sollte folgenden Eintrag enthalten bzw. um diese ergänzt werden:

```
wrapper.java.classpath.N=lib\extensions\qs-reports-<version>\*.jar
```



Dabei ist „N“ variabel und sollte eine fortlaufende Zahl der vorhergehenden Einträge sein!



<version> ist dabei die Version von Reports für Intrexx

Die Datei *log4j.properties* sollte um die folgende Zeile erweitert werden:

```
log4j.logger.de.quinscape=INFO
```

Für den Betrieb auf Unix/Linux Systemen, sollte zusätzlich noch folgender Eintrag ergänzt werden:

```
wrapper.java.additional.N=-Djava.awt.headless=true
```




Dabei ist „N“ variabel und sollte eine fortlaufende Zahl der vorhergehenden Einträge sein!

3.3 Probleme nach einem Update von Intrexx

Bitte beachten Sie, dass für verschiedene Versionen von Intrexx auch passende Versionen von „Reports für Intrexx“ existieren. Diese Reports für Intrexx Versionen sind an die jeweilige Intrexx Version gebunden und funktionieren nicht mit anderen Intrexx Versionen. Eine detaillierte Übersicht der Versionen finden Sie im Anhang A.1.

Sollten Sie ein Portal mit „Reports für Intrexx“ in eine inkompatible Version von Intrexx importiert haben, dann kann es sein, dass dieses Portal nicht mehr startet. Sofern sie noch kein passendes Update der Version von „Reports für Intrexx“ durchführen können, hilft der folgende „work around“.

Benennen Sie die Datei *quinscape-reports-context.xml* im Portalverzeichnis unter  *internal\cfg\spring* um, bspw. in *quinscape-reports-context.xml.bak*. Das Portal startet dann problemlos, aber Reports für Intrexx kann dann natürlich nicht genutzt werden.

3.4 Entfernen alter Bibliotheken, Reports für Intrexx 8.0.1




Um die folgenden Schritte durchführen zu können, müssen Sie zunächst den Portaldienst beenden.

Falls Sie von der Version 8.0.1 aktualisieren, löschen Sie bitte das Verzeichnis




<portal>\lib\extensions\qs-reports-8.0.1

und überprüfen Sie die Konfigurationsdatei *portal.wcf* im Verzeichnis  `<portal>\internal\cfg`. Aus dieser Datei müssen sämtliche Einträge entfernt werden, die wie Folgt beginnen:

```
wrapper.java.classpath.N=<portal>\lib\extensions\qs-reports-8.0.1\
```



Dabei ist „N“ variabel und ist eine fortlaufende Zahl der vorhergehenden Einträge!

Entfernen Sie zudem die Datei *quinscape-reports-context.xml* im Portalverzeichnis unter  `internal\cfg\spring`.

Anschließend führen Sie den Import des Anwendungspakets gemäß Abschnitt 3.6 durch.

3.5 Update von einer älteren Version

In diesem Handbuch werden nur die notwendigen Schritte für Updates von der Version Reports für Intrexx 8.0 oder höheren Versionen beschrieben. Für Updates älterer Versionen kontaktieren Sie bitte unseren Support.

3.6 Import des Anwendungspaket durchführen



Achtung: Sofern Sie das Portal nach den vorherigen Schritten noch nicht wieder gestartet haben, so müssen Sie dies für den folgenden Import erledigen.

Die Applikation Reports für Intrexx dient der Verwaltung Ihrer Berichte und deren Veröffentlichung im Portal.

Importieren Sie die Datei *Intrexx_Reports_Applikation_*.zip*. Diese Datei befindet sich im Installationspaket. Dabei handelt es sich um ein seit der Intrexx Version 5.1 verfügbares Anwendungspaket, das alle notwendigen Punkte (z.B. die Applikation, Prozesse, Konfigurationen, benötigte Bibliotheken) enthält. Hier wählen Sie dann die oben genannte Datei aus.

Das Paket besteht aus der Applikation Reports für Intrexx, den Reports für Intrexx-Prozessen, Konfigurationsdateien und einigen Bibliotheken. Wählen Sie im nächsten Schritt („Weiter“) die Option „Bestehende Applikationen überschreiben“ bei einem Update und bestätigen Sie das Überschreiben der Applikation. Anschließend klicken Sie auf „Importieren“.

Bei einer **Neuinstallation** werden Sie aufgefordert, Ihren **Lizenzschlüssel** für Reports für Intrexx einzugeben. Den Lizenzschlüssel haben Sie in den Ihnen zugestellten Unterlagen erhalten. Sofern Sie Reports für Intrexx in einem weiteren Portal einsetzen wollen, nehmen Sie bitte Kontakt mit uns auf: reports@quinscape.de. Nachdem Sie Ihren Lizenzschlüssel erfolgreich eingegeben haben, wird die Applikation importiert.

Als nächstes werden die im Applikationspaket enthaltenen Konfigurationsdateien importiert. In diesem Dialog „2. Konfigurationsdateien“ übernehmen Sie alle Dateien, sofern

sie nicht durch andere Plugins vorhanden sind und bestätigen auch diesen Dialog mit „Weiter“. Anschließend klicken Sie auf „Importieren“.

Der nächste Schritt ist der Import der im Applikationspaket enthaltenen Reports für Intrexx-Prozesse. Diese sind für den reibungslosen Ablauf von Reports für Intrexx zwingend erforderlich! Bestätigen diesen Dialog mit „Weiter“ und klicken Sie anschließend auf „Importieren“.



Hinweis: Im Falle eines Updates auch hier bitte „Bestehende Prozesse überschreiben“ wählen.

Abschließend können Sie den Veröffentlichungsort der Applikation Reports für Intrexx festlegen. Standardmäßig wird dabei der Menüordner System vorgeschlagen.

Für den Normalbetrieb müssen keine speziellen Berechtigungen eingestellt werden. Ein Zugriff der Anwendung sollte administrativen Benutzern vorbehalten sein.

Im Nachgang an den Import des Anwendungspakets muss der in diesem Rahmen importierte Prozess *Reports für Intrexx – Setup* einmalig ausgeführt werden. Dieser kopiert noch einige notwendige Dateien (Skripte, Beispiele, etc.) an die passenden Stellen im Portalverzeichnis.


Diese Ausführung können Sie bspw. im Portalmanager in der Aufgabenplanung unter Werkzeuge einmalig durchführen. An dieser Stelle muss die Aufgabe „Globaler Prozess-Timer (Reports für Intrexx – Setup, Import des Pakets abgeschlossen, ...)“ via Rechtsklick und der Auswahl von Starten gestartet werden.

Den Hinweis auf den Portalneustart ignorieren Sie bitte an dieser Stelle, sondern fahren erst mit dem nächsten Abschnitt – der Überprüfung der Konfigurationsdateien – fort.

3.7 Überprüfung der Konfigurationsdateien

Überprüfen Sie die Konfigurationsdatei *log4j.properties* im Verzeichnis  *<portal>internal/cfg*. Diese Datei sollte um die folgende Zeile erweitert werden:

```
log4j.logger.de.quinscape=INFO
```

Überprüfen Sie die Konfigurationsdatei *portal.wcf* im Verzeichnis  *<portal>internal/cfg*. Für den Betrieb auf Unix/Linux Systemen, sollte folgender Eintrag ergänzt werden:

```
wrapper.java.additional.N=-Djava.awt.headless=true
```



Dabei ist „N“ variabel und sollte eine fortlaufende Zahl der vorhergehenden Einträge sein!

Nun müssen Sie Ihr Portal neustarten.

3.8 Migrationsprozess ausführen

Falls Sie keine Neuinstallation von Reports für Intrexx ausführen, sondern Ihre bestehende Installation aktualisieren, müssen Sie nach dem gerade erfolgten Portal-Neustart den Prozess „Reports für Intrexx – Migration“ in gleicher Weise ausführen, wie Sie es mit dem Setup-Prozess getan haben.

Der Migrationsprozess sorgt nach dem Portal-Neustart dafür, dass Ihre bestehenden Reports auf das jeweils aktuelle Format von Reports für Intrexx gebracht werden.

3.9 Konfiguration von Intrexx unter Linux

Folgende Anpassungen sind nötig, damit JasperReports unter Linux läuft:

3.9.1 Java im headless-Modus starten

In der Datei `internal/cfg/portal.wcf` innerhalb Ihres Intrexx-Portals ist ein weiterer Parameter hinzuzufügen. Er lautet

```
wrapper.java.additional.N=-headless=true
```

wobei **N** die nächste freie Nummer in der Reihe der schon vorhandenen additional-Zeilen ist.

3.9.2 Hinzufügen einer fontconfig.properties

Im `lib`-Verzeichnis der Java-Installation (zum Beispiel `/opt/intrexx/jre`) ist eine `fontconfig.properties` hinzuzufügen, falls noch nicht vorhanden. Diese Datei konfiguriert die für Java verfügbaren Schriftarten.

3.10 Reports für Intrexx – Beispieldaten und Prozess

Die Applikation *Reports für Intrexx – Beispieldaten* enthält Beispiele für die verschiedenen Möglichkeiten, wie Berichte aus Ihren Anwendungen/Ihrem Portal heraus erzeugt werden können. Die Applikation besteht zum einen aus der Applikation, die mit Daten importiert werden sollte, sowie aus dem Prozess *Reportdemo - Groovy action*. Sie finden

beide im Applikationspaket „Reportbeispieldaten und -prozess.zip“ im Verzeichnis `<portal>\quinscape\samples`. Dieses importieren Sie analog zum Applikationspaket für Reports für Intrexx. Achten Sie dabei darauf, dass Sie die Applikation „Mit Daten“ importieren – sonst sind die Beispieldaten nicht vorhanden.



4 Deinstallation

Um Reports für Intrexx zu deinstallieren, folgen Sie bitte diesen Schritten. Beachten Sie bitte, dass dies sämtliche in Reports für Intrexx eingepflegten Berichte und Ressourcen entfernen wird; Sie sollten eine Sicherungskopie anlegen, indem Sie die Applikation Reports für Intrexx samt Daten exportieren.

1. Nehmen Sie die o. g. Sicherung vor, falls gewünscht.
2. Führen Sie den Prozess *Reports für Intrexx – Deinstallation* aus.
3. Löschen Sie alle Prozesse, die mit „Reports für Intrexx – ...“ beginnen.
4. Löschen Sie alle Applikationen, die mit „Reports für Intrexx – ...“ beginnen.
5. Starten Sie Ihr Portal neu.
6. Löschen Sie das Verzeichnis
`<portal>/lib/extensions/qs-reports-19.03*`.

5 Ein erster Bericht

Dieses Kapitel beschreibt die Schritte zu Ihrem ersten eigenen Bericht.

5.1 Ziel

Ziel des Berichts soll eine Mitarbeiterliste sein, die per PDF aus dem Portal heraus aufgerufen werden kann.

5.2 Berichtserstellung mit dem Jaspersoft Studio

Reports für Intrexx baut technologisch auf der Reporting Engine JasperReports und dem dazugehörigen graphischen Editor dem Jaspersoft Studio auf. Im Folgenden wird zunächst die Einrichtung des Editors beschrieben, danach wird auf die Erstellung des Berichts eingegangen.



Hinweis: Das Jaspersoft Studio ist ein auf Eclipse basierender Berichtsdesigner für JasperReports und den JasperReports Server. Das Jaspersoft Studio ist eine komplette Neufassung des iReport Designers, welches sowohl als Eclipse Plug-In als auch als Stand-Alone-Anwendung verfügbar ist.

5.2.1 Installation und Konfiguration


Das Jaspersoft Studio wird mit Installern für Windows, Linux und Mac OS X ausgeliefert. Im Installationspaket erhalten Sie einen Downloadlink für die zum Zeitpunkt dieses Releases gültige Version Jaspersoft Studio 6.9.0¹. Die Versionen sind bisher abwärtskompatibel. Das heißt Berichte aus vorherigen Versionen von iReport und dem Jaspersoft Studio können in neuen Versionen von iReport und dem Jaspersoft Studio geöffnet werden. Andersherum gilt dies nicht zwingend. Beachten Sie bitte darüber hinaus, dass Reports für Intrexx immer für eine bestimmte Version von JasperReports bzw. dem Jaspersoft Studio entwickelt wurde; genauere Informationen können Sie Anhang A.1 entnehmen. Wenn Sie eine neuere Version verwenden möchten, dann müssen Sie im Jaspersoft Studio den entsprechenden Kompatibilitätsmodus einstellen. Laden Sie die für Sie korrekte Version (Kompatibilität und Betriebssystem) herunter, führen Sie den Installer aus und folgen Sie den darin enthaltenen Anweisungen.



Hinweis: Das Jaspersoft Studio kann, muss aber nicht auf dem Server installiert werden. Wichtig ist, dass der Rechner auf dem das Jaspersoft Studio installiert wird auf die Portal-Datenbank (und ggf. weitere zu verwendende Datenbanken Zugriff

¹ Hinweis: Die im Folgenden aufgeführten Screenshots zeigen z.T. eine ältere Version des Jaspersoft Studios. Dies ist kein Mangel, da sich die Oberfläche in den größten Teilen nicht verändert hat.

hat). Der Rechner, auf dem das Jaspersoft Studio installiert wird, benötigt ein JRE² Version 1.8 oder höher.

 **Hinweis:** In den Beschreibungen und Screenshots im Folgenden wird davon ausgegangen, dass Sie das Jaspersoft Studio in der Stand-Alone-Variante installiert haben.

Nach erfolgreicher Installation starten Sie das Programm.

Sofern Sie das Jaspersoft Studio zum ersten Mal starten, müssen Sie zunächst einen Workspace definieren.

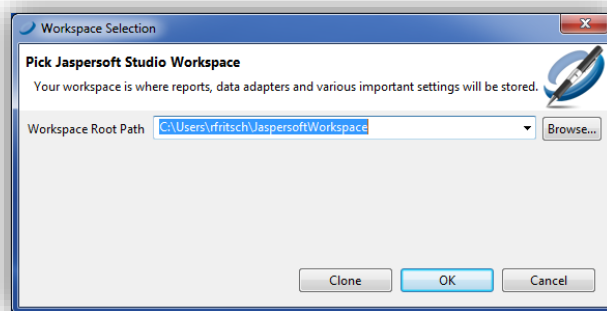



Abbildung 5 Workspace für das Jaspersoft Studio definieren

Hierfür müssen Sie einen Pfad hinterlegen, in dem die Einstellungen und Ressourcen des Workspace abgespeichert werden sollen. Falls das Verzeichnis in den Sie den Workspace anlegen möchten noch nicht existiert, können Sie es automatisch über  **OK** durch das Jaspersoft Studio anlegen lassen.

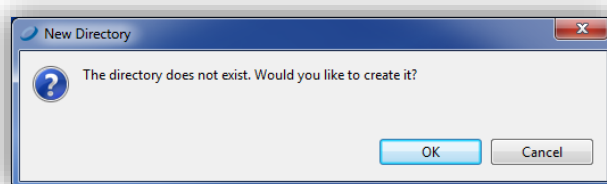



Abbildung 6 Verzeichnis für den Workspace durch das Jaspersoft Studio erstellen lassen

Das Jaspersoft Studio startet nun und beginnt mit einem Willkommensdialog.

² Java Runtime Edition (zu beziehen unter  <https://java.com/de/download/>)

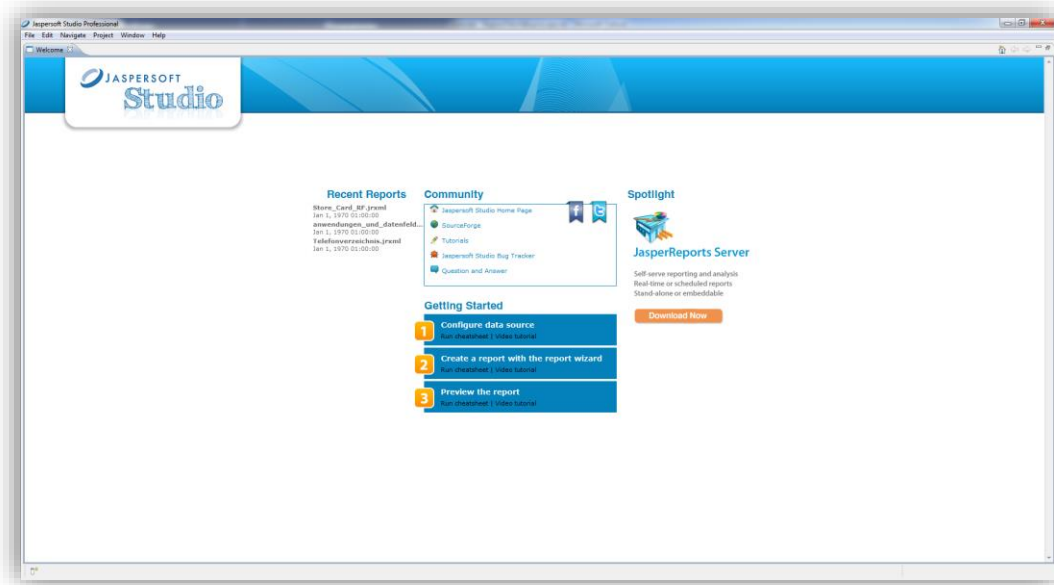





Abbildung 7 Jaspersoft Studio Startseite

Als nächstes müssen Sie ein Projekt anlegen. In diesem Projekt können Sie dann sämtliche Reports, Subreports und weitere Ressourcen wie z. B. Bilder verwalten.

 **Hinweis:** Die Sicht „Project Explorer“ muss für die Einrichtung zur Verfügung stehen. Diese kann bei Bedarf über das Hauptmenü  *Window, Show View* geöffnet werden.

Über das  *File, New, Project* können Sie ein neues Projekt erstellen. Erstellen Sie ein neues *Jaspersoft Studio, JasperReports Project*.

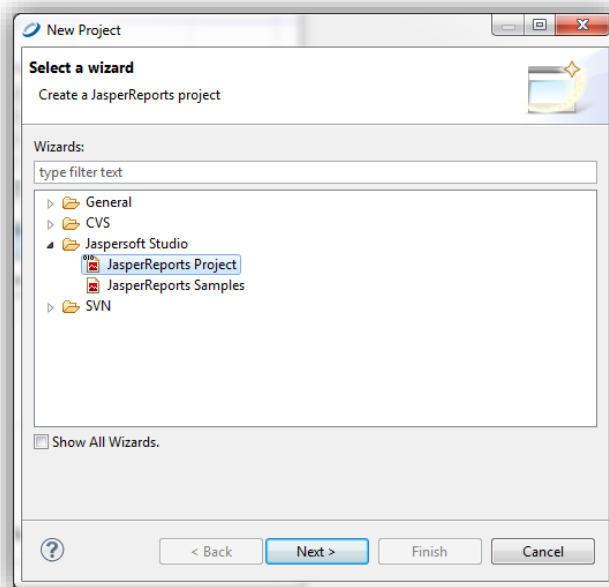








Abbildung 8 Erstellung eines JasperReports Projekts im Jaspersoft Studio

Über  *Next* können Sie im nächsten Schritt einen Namen für das Projekt vergeben und über  *Finish* das Projekt erstellen.

Als nächstes muss eine Datenbankverbindung zu Ihrem Intrexx Portal hergestellt werden. Je nach der von Ihnen gewählten Portal-Datenbank benötigen Sie einen entsprechenden Datenbanktreiber. Diesen finden Sie bspw. im  *<intrexx>lib* Verzeichnis Ihrer Intrexx Installation bzw. auf der Intrexx-Installations-CD.

Im Jaspersoft Studio werden zur Anbindung der Datenbank die sogenannten Data Adapter benutzt.

 **Hinweis:** Die Sicht „Repository Explorer“ muss für die Einrichtung zur Verfügung stehen. Diese kann bei Bedarf über das Hauptmenü  *Window, Show View* geöffnet werden.

Fügen Sie im *Repository Explorer* unter dem Knoten Data Adapters über  *Create Data Adapter* einen neuen Data Adapter hinzu.

Im folgenden Dialog werden Sie aufgefordert den Typ des Data Adapters auszuwählen. Wählen Sie dazu den passenden Adapter aus. In der folgenden Beschreibung gehen wir von einer *Database JDBC Connection* aus.

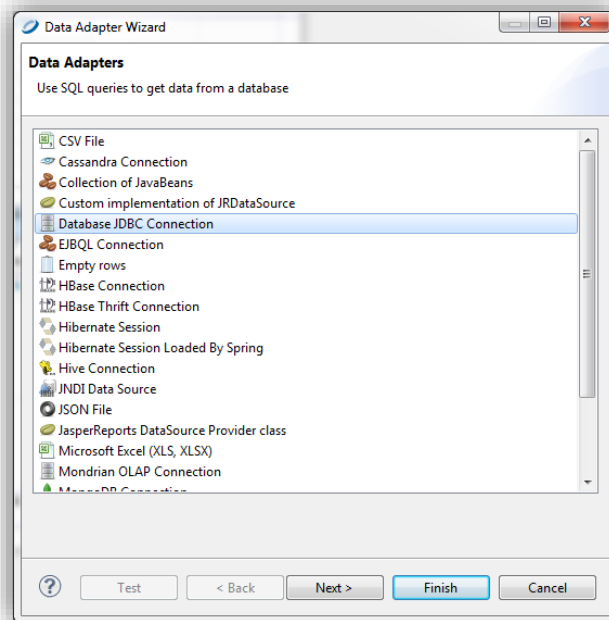



Abbildung 9 Erstellung einer Database JDBC Connection

Nach einem Klick auf  *Next* können Sie die notwendigen Einstellungen für eine Verbindung mit Ihrer Datenbank vornehmen.

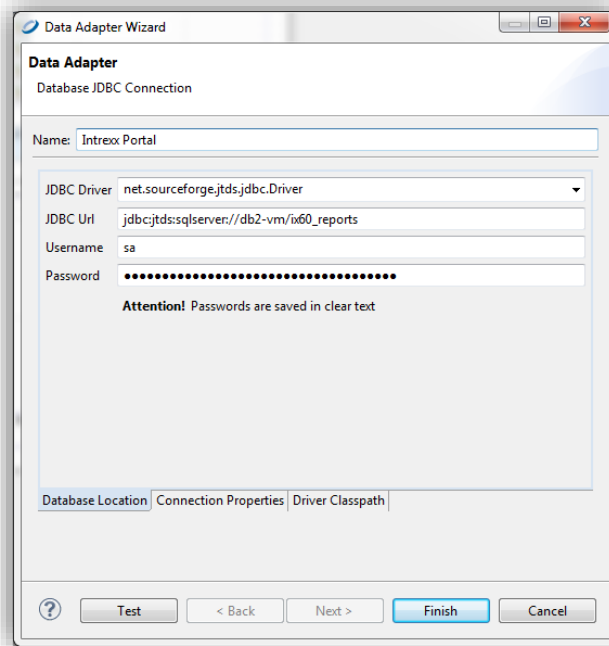



Abbildung 10 Exemplarische Datenbankverbindung

Über  *Finish* können Sie den Data Adapter abspeichern.

Die notwendigen Informationen können Sie u.a. aus dem Eigenschaftendialog der Intrex Systemdatenquelle (IxSysDb) im *Bereichs Integration/Datenquellen* Ihres *Intrex Portal Managers* ablesen.

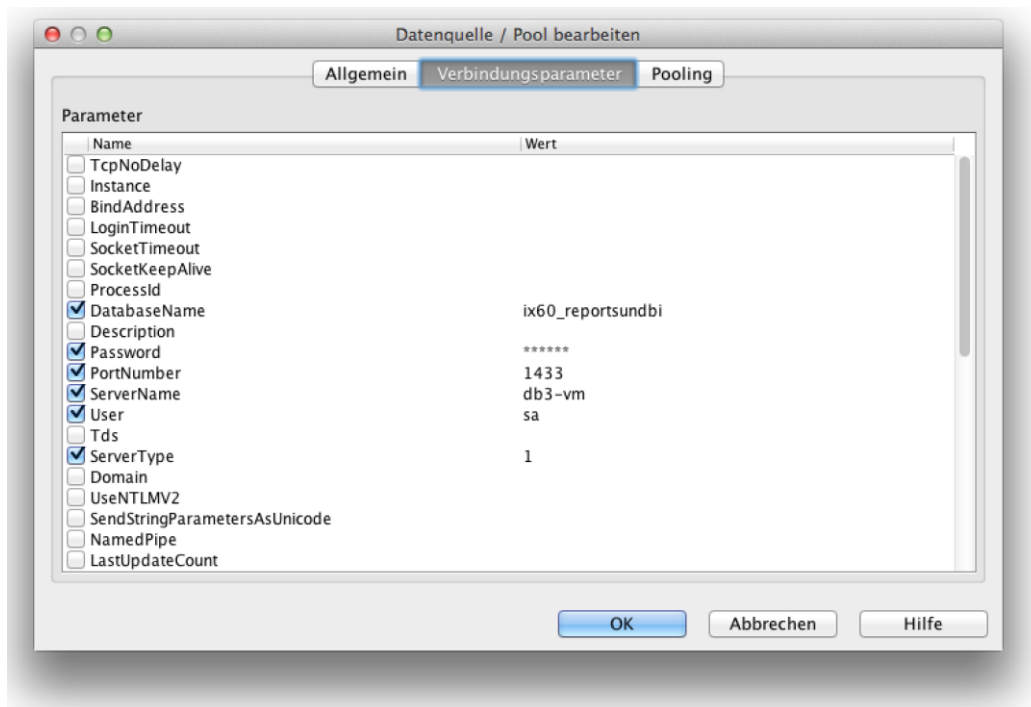


Abbildung 11 Bestimmung der JDBC Verbindungseigenschaften aus dem Intrexx Portal Manager

Nach erfolgter Einrichtung wird die eingestellte Verbindung im Bereich *Repository Explorer* unterhalb von *Data Adapters* aufgeführt.

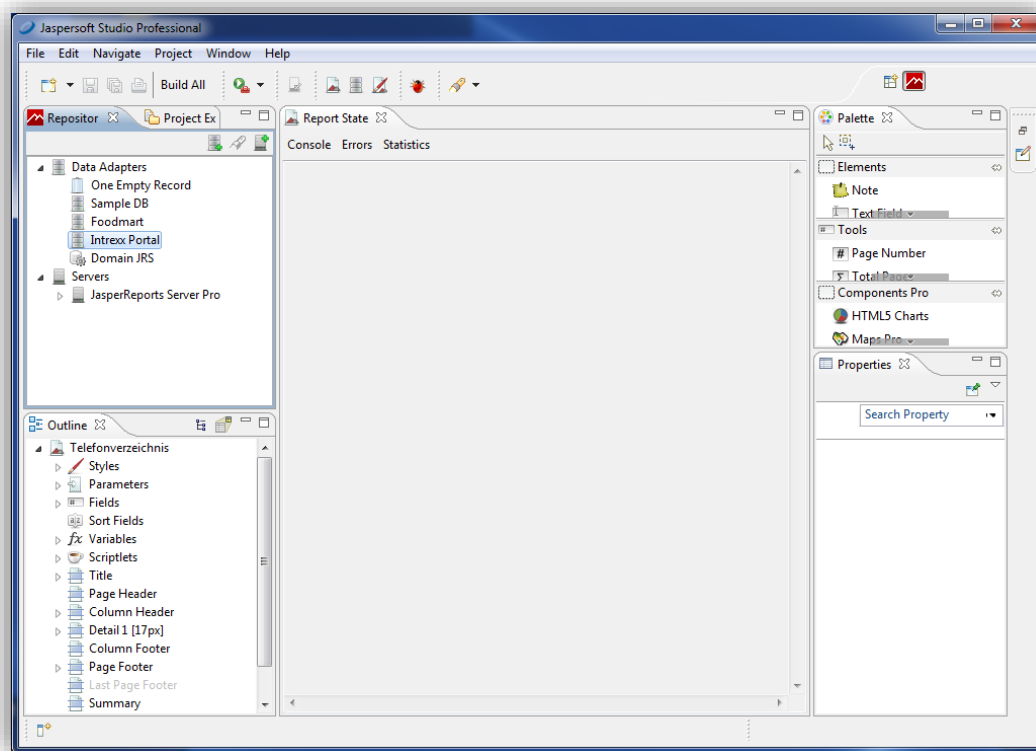


Abbildung 12 Datenbankeinrichtung erfolgt

Nun können wir mit einem Bericht beginnen. Den Bereich Repository Explorer können Sie wieder schließen.

Erstellen Sie dann mittels des Menüpunkts  *File, New, Jasper Report* einen neuen Bericht mit dem Assistenten. Sie können dabei aus einer Reihe von Vorlagen wählen.

Klicken Sie dann auf  *Next*.

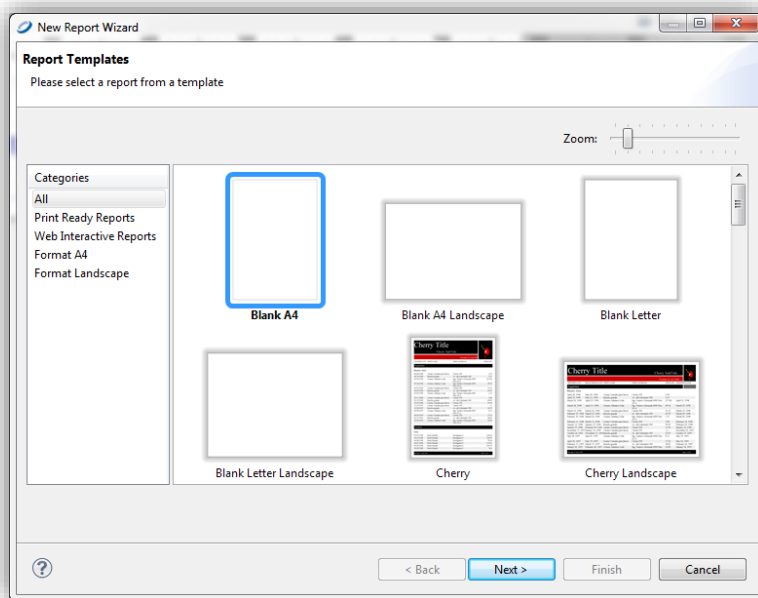


Abbildung 13 Dialog zur Erstellung eines neuen Berichts

In den nachfolgenden Schritten werden Sie durch die notwendigen Einstellungen für einen ersten Bericht geführt. Wählen Sie das Projekt in das der Bericht gespeichert werden soll, geben Sie einen Namen für den Bericht an, definieren Sie dann die passende Datenbankabfrage (z.B. `select * from vbluser`, um die Liste aller Intrex-Benutzer Ihres Portals zu erhalten), definieren Sie anschließend die Felder, die ausgelesen werden sollen und geben abschließend eine optionale Gruppierung an, nach der die Daten gruppiert werden sollen³.

³ Dann müssen die Daten auch danach sortiert werden.

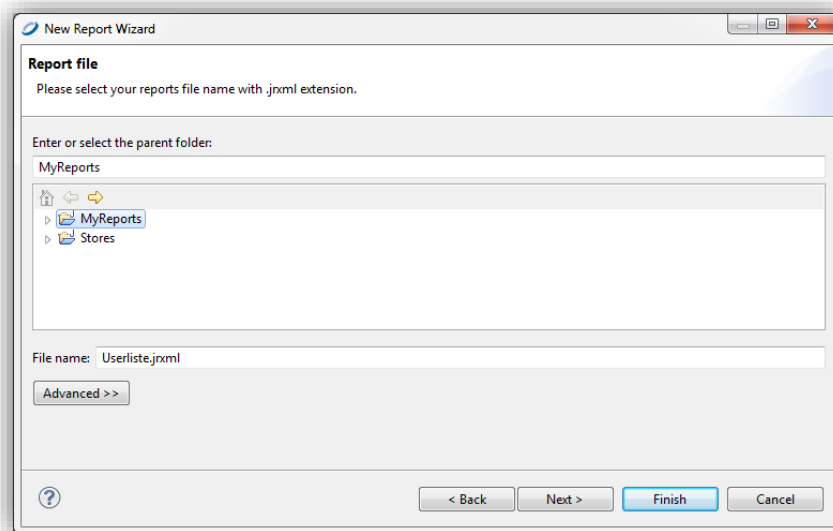


Abbildung 14 Berichtsname und Projekt

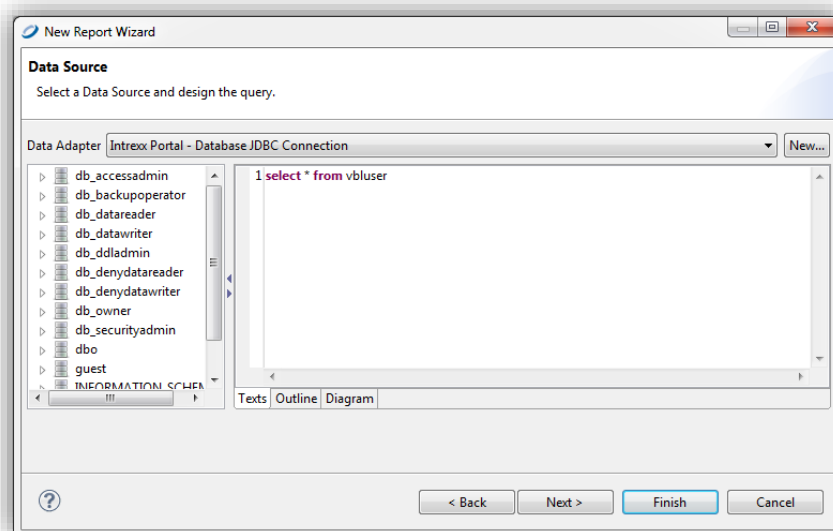


Abbildung 15 Datenbankabfrage

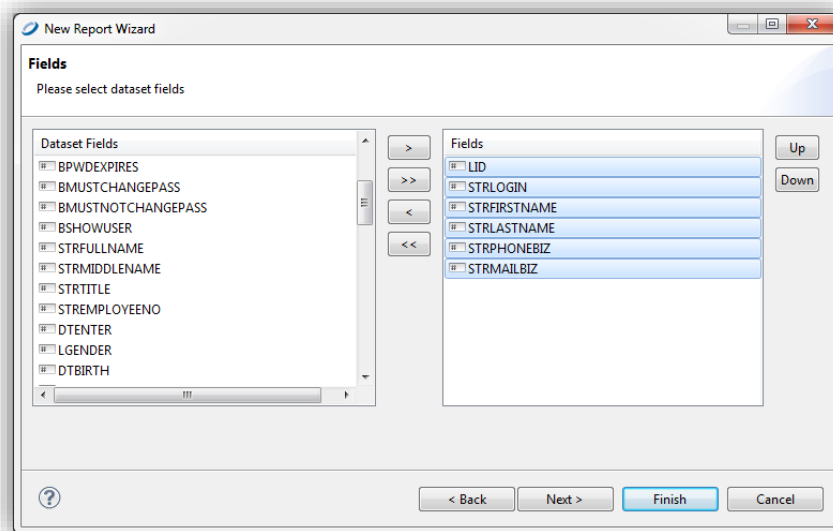


Abbildung 16 Auswahl der Felder

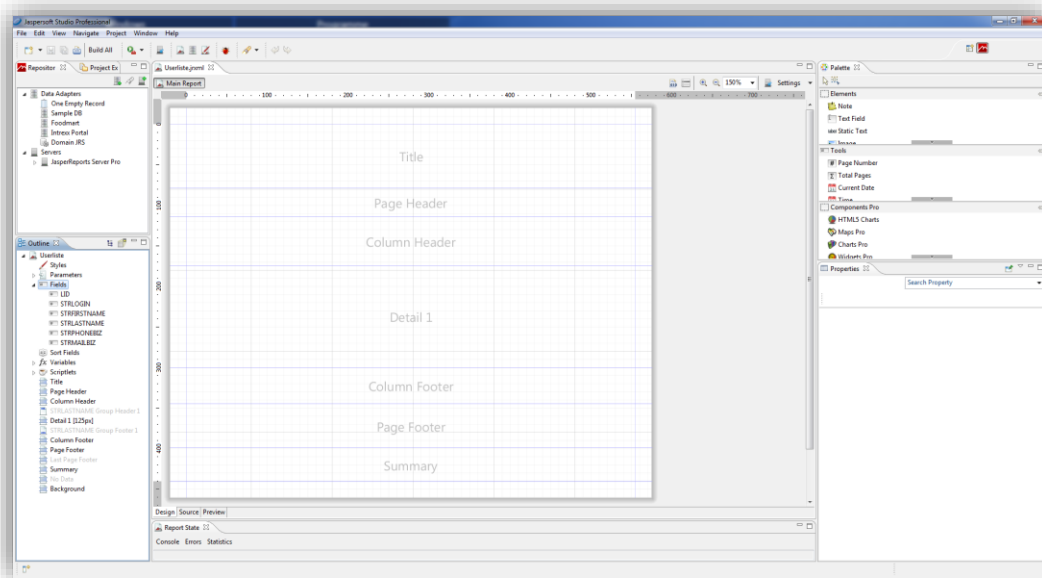



Abbildung 17 Initiale Darstellung eines neuen Reports mit aufgeklappter Feldliste

Fügen Sie mittels des *Static Text* einen Seitentitel im Bereich Titel und Spaltenüberschriften ein.  Das Element finden Sie in der Sicht *Palette*.

Ziehen Sie dann sukzessive die jeweiligen Felder auf den Bericht und richten Sie sie anhand der von Ihnen definierten Spalten aus.

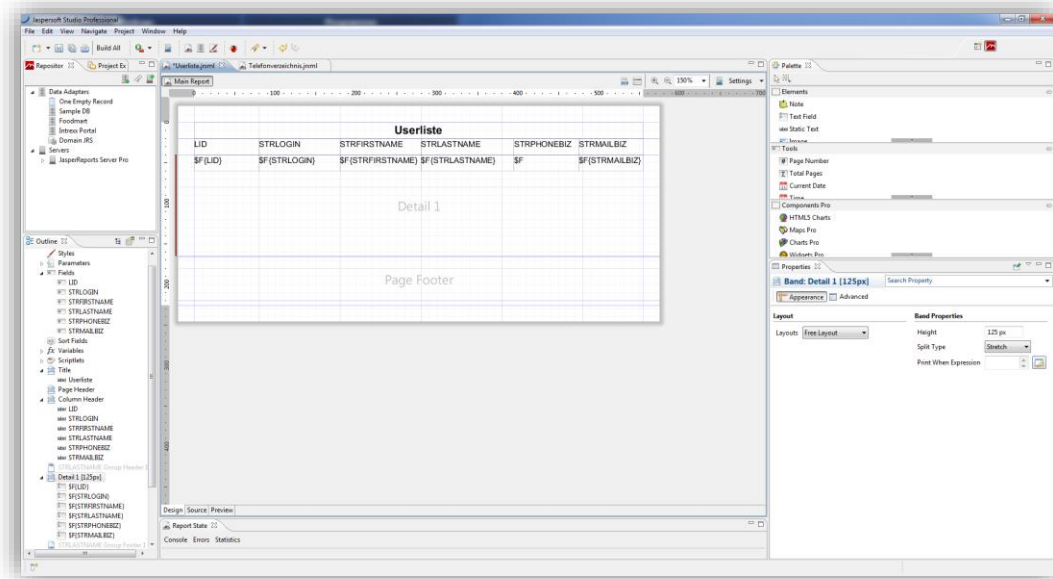


Abbildung 18 Bericht mit Datenbankfeldern

Abschließend fügen Sie noch ein *Seite x von y* Element (in der *Palette* unter *Tools*) in den Bereich *Page Footer*.

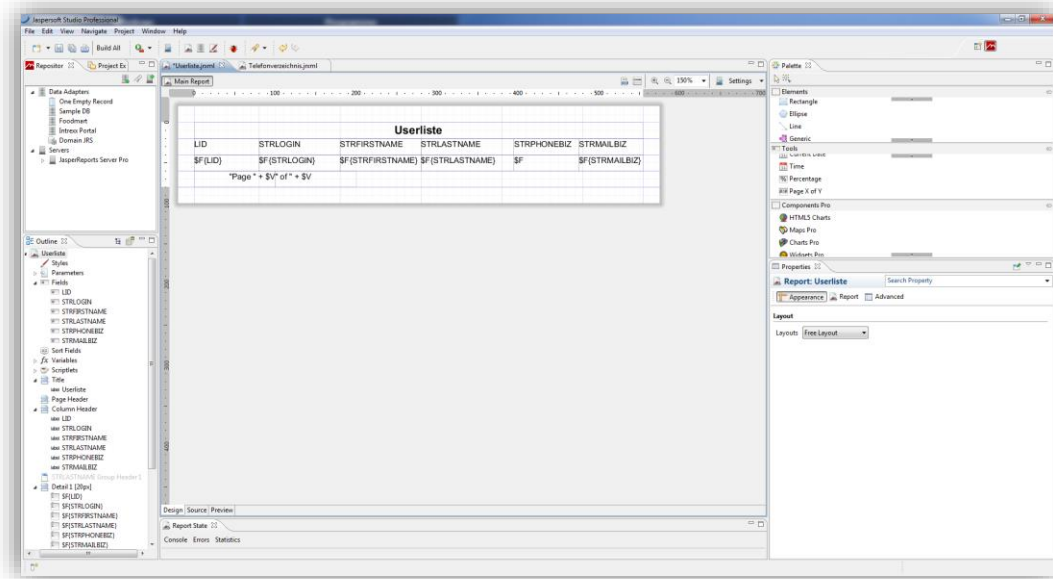


Abbildung 19 Fertiger Bericht

Mittels eines eingebauten *Preview* Modus können Sie dann sehen, wie der Bericht später aussieht.

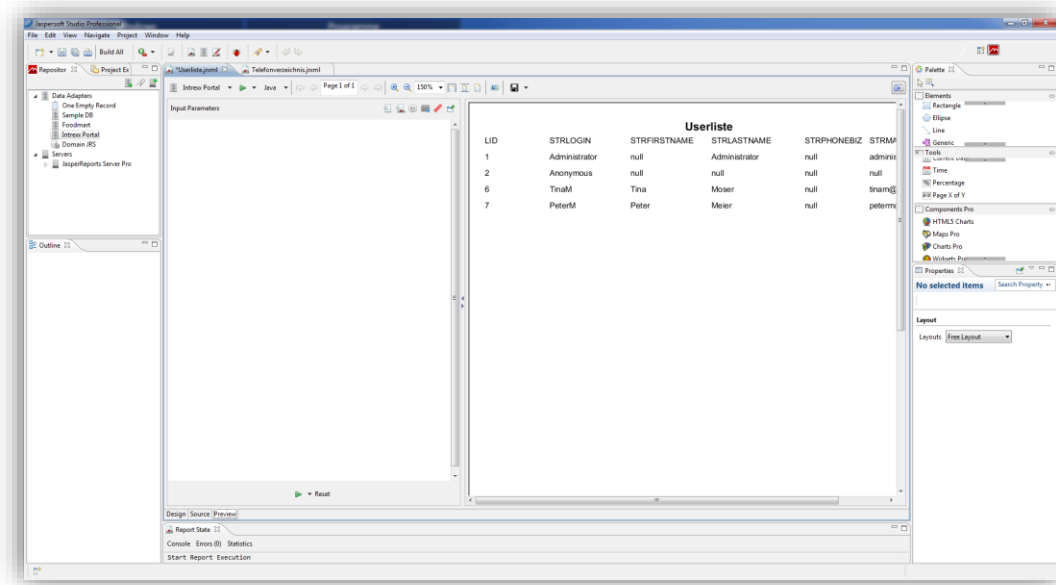


Abbildung 20 Berichtsvorschau

Es ist sicherlich noch etwas Feinschliff nötig, aber im Wesentlichen sind die Informationen enthalten, die gefordert waren. Weiterführende Informationen zur Berichtserstellung finden Sie unter anderem in der JasperSoft Studio/JasperReports-Dokumentation (siehe 6.3) oder im Rahmen unseres Schulungsangebots (siehe 6.1)

5.3 Veröffentlichung des Berichts im Portal

Damit Ihr Bericht aus dem Portal heraus aufgerufen werden kann, muss dieser in der Applikation *Reports für Intrexx* veröffentlicht werden. Rufen Sie dazu Ihr Portal im Browser auf und wechseln Sie in die Applikation.



Abbildung 21 Startseite der Applikation Reports für Intrexx

Die Anwendung Reports für Intrexx enthält eine Übersicht der bislang veröffentlichten Berichte. Diese Übersicht kann über das Applikationsmenü *Berichte* oder über die Schnellstartfunktion *Alle Berichte* auf der Startseite der Applikation aufgerufen werden. Je nachdem, ob Sie die Anwendung mit oder ohne Beispieldaten (siehe Kapitel 3.6) importiert haben, sind Berichte in der Tabelle enthalten oder nicht.

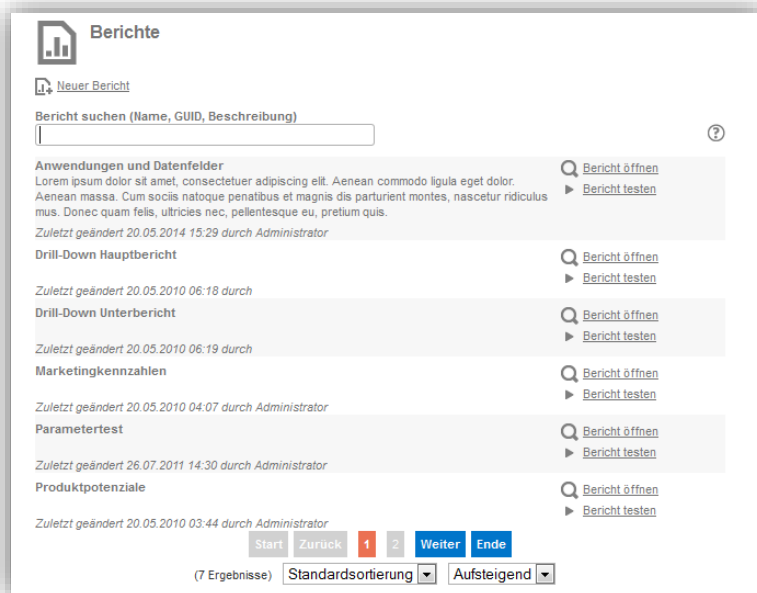


Abbildung 22 Berichtsübersicht der Applikation *Reports für Intrexx*



Klicken Sie auf die Schaltfläche  **Neuer Bericht**. Der Berichtserstellungs-Assistent führt Sie schrittweise durch die Erstellung eines neuen Berichts.



Abbildung 23 Berichtsassistent Schritt 1 - Definition eines Berichts

Im ersten Schritt legen Sie einen Bericht an. Vergeben Sie einen sprechenden Namen. Anhand dieses Namens können Sie den Bericht aus Ihrer Applikation später aufrufen – daher muss er eindeutig sein. Alternativ können Sie Ihre Berichte auch über die

automatisch vergebene GUID aufrufen. Die Verteilerkontrolle Berechtigte legt fest, welche Benutzer über das Reports für Intrexx-Servlet den Bericht aufrufen können (siehe auch Kapitel 5.5.2). Klicken Sie auf die Schaltfläche  *Weiter*, um zum zweiten Schritt des Assistenten zu gelangen.

Im zweiten Schritt definieren Sie die Ressourcen des Berichtes. In einem Bericht werden die einzelnen Ressourcen zusammengefasst. In unserem Beispiel besteht der Bericht aus nur einer Ressource. Komplexere Berichte haben durchaus eine Vielzahl von Ressourcen.


 Ressourcen können alternativ auch unabhängig von Berichten über das Applikationsmenü *Ressourcen* in der Applikation hinterlegt und verfügbar gemacht werden.



Abbildung 24 Berichtsassistent Schritt 2 - Ressourcenzuordnung



Klicken Sie auf die Schaltfläche  *Neue Ressource*. Es öffnet sich ein Tooltip, in dem Sie nach Bezeichnung und der Ressource selbst gefragt werden. Unsere Ressource für diesen Bericht ist die soeben erzeugte Berichtsdefinition *Userliste*. Geben Sie eine sprechende Bezeichnung für Ressource an. Es empfiehlt sich, den Namen der Berichtsdefinition zu verwenden. Als Ressource selbst wählen Sie die gespeicherte Datei der Berichtsdefinition - die XML-Definitionsdatei *.jrxml*. Diese wird server-seitig automatisch kompiliert und die *.jrxml* Datei wird durch die passende *.jasper* Datei ersetzt. Dadurch wird sichergestellt, dass die in der Beschreibungs-Datei verwendeten Komponenten mit der in Intrexx verfügbaren Version kompiliert werden können.



Abbildung 25 Ressource erfassen

Im selben Schritt können Sie Ihrem Bericht Ressourcen zuordnen. Diese Zuordnung bestimmt, auf welche Ressourcen Ihr Bericht zugreifen kann. Klicken Sie dazu auf  *Ressource zuordnen*.

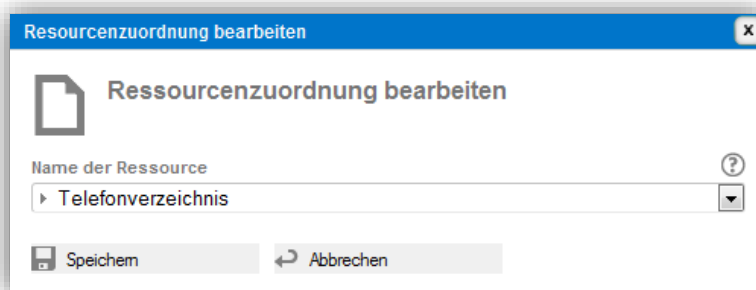



Abbildung 26 Ressource auswählen

Klicken Sie anschließend auf die Schaltfläche  *Weiter*, um zum dritten Schritt des Assistenten zu gelangen und die Hauptberichts-Ressource zuzuordnen. Der Hauptbericht ist diejenige Ressource, die als erste in der Berichtsbearbeitung aufgerufen wird, d.h. der Einsprung in den Bericht. Es gibt für einen Bericht genau eine Hauptberichts-Ressource, der aus der Menge der dem Bericht zugeordneten Ressourcen stammt. Bitte wählen Sie hier eine .jasper Ressource.

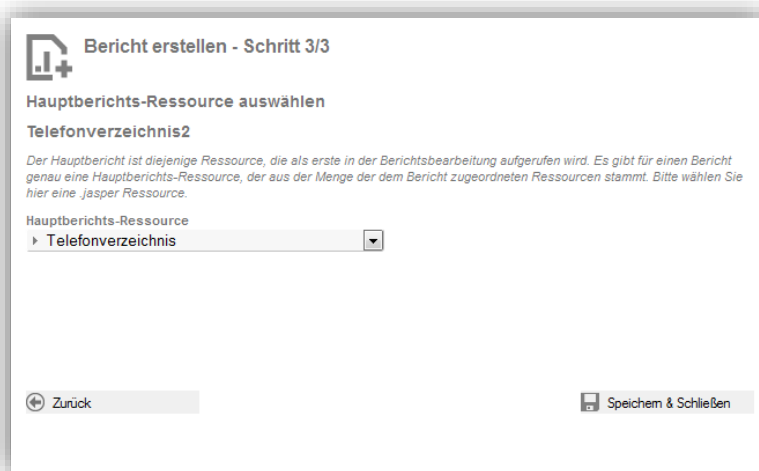




Abbildung 27 Hauptberichts-Ressource auswählen

Durch Klick auf  *Speichern & Schließen* beenden Sie den Assistenten und werden auf die Berichtsübersicht geleitet. Von dort können Sie den Bericht über die  *Bericht testen* Schaltfläche erzeugen.

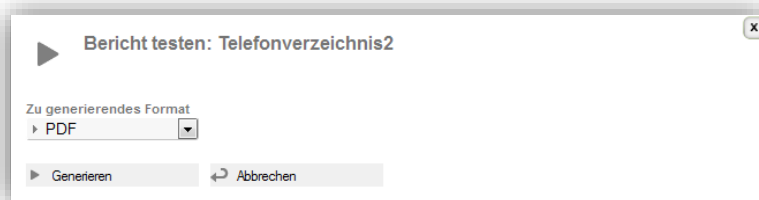



Abbildung 28 Bericht testen

5.4 Pakete

Seit der Reports für Intrexx Version 6.0.3 werden die sogenannten Pakete zur Verfügung gestellt. Pakete sind eine Zusammenstellung von Berichten und ihren zugehörigen Ressourcen. Über die Paketfunktionalität können Berichte beispielsweise einfach versioniert oder auf ein weiteres System übertragen werden.

Eine Übersicht der Pakete kann über das Applikationsmenü  *Pakete* aufgerufen werden. Dort wird eine Liste der Pakete, die Sie in Ihrem Portal importiert oder exportiert haben dargestellt.

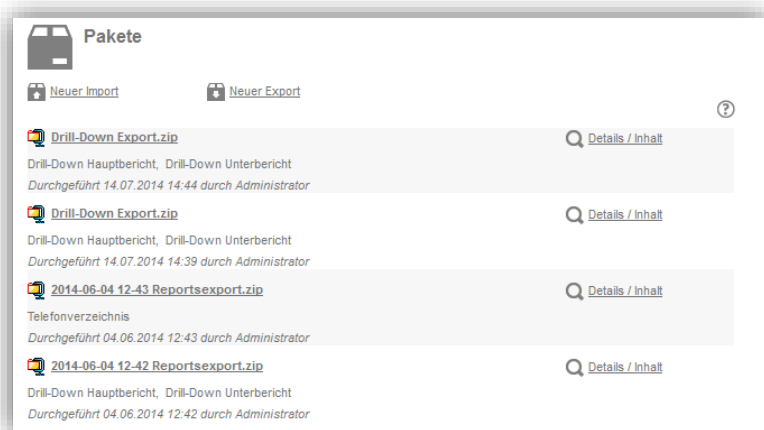




Abbildung 29 Übersicht der Pakete

Über die Schaltfläche  *Details / Inhalt* können Sie sich die Inhalte des Pakets anzeigen lassen. Des Weiteren wird der Name des Pakets, das Durchführungsdatum und durch welchen Benutzer die Aktion erfolgte, angezeigt.

5.4.1 Export

Über die Schaltfläche  *Neuer Export* wird der Assistent für den Berichtsexport aufgerufen. Die Verteilerkontrolle legt die zu exportierenden Berichte fest.

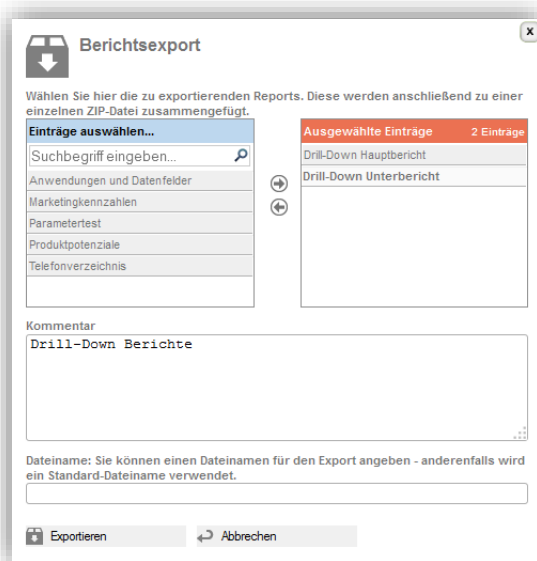




Abbildung 30 Assistent für den Berichtsexport

Durch einen Klick auf die Schaltfläche  *Exportieren* wird das Export-Paket zusammengestellt und angezeigt. Sie können nun die zusammengestellte ZIP-Datei mit den Berichten herunterladen, um sie in einem anderen Portal zu importieren.

Ein Klick auf die Schaltfläche  *Ergebnis löschen* verwirft das Paket; wählen Sie nur  *Schließen* so wird das Paket in der Übersicht der Pakete aufgeführt.

5.4.2 Import



Über die Schaltfläche  *Neuer Import* wird der Assistent für den Berichtsimport aufgerufen. Im ersten Schritt muss die Importdatei (ZIP-Datei) ausgewählt werden, welche die zu importierenden Reports enthält.



Abbildung 31 Importpaket auswählen

Durch einen Klick auf die Schaltfläche  *Einlesen* wird Ihnen anschließend angezeigt welche Berichte in der Datei enthalten sind. Es wird noch kein Import der Berichte durchgeführt.

Im folgenden Schritt wählen Sie aus, welche der enthaltenen Reports Sie importieren möchten.




 **Achtung:** Jeder vorhandene Report wird beim Import inkl. der Ressourcen überschrieben - daher ist ein vorheriger Export der Berichte sinnvoll.



Abbildung 32 Zu importierende Reports auswählen

Durch einen Klick auf die Schaltfläche  *Importieren* werden die ausgewählten Berichte samt Ressourcen importiert.

Ein Klick auf die Schaltfläche  *Ergebnis löschen* verwirft das Paket; wählen Sie nur  *Schließen* so wird das Paket in der Übersicht der Pakete aufgeführt.

5.5 Aufruf des Berichts aus Ihrer eigenen Anwendung

Seine Stärken spielt Reports für Intrexx aus, wenn Ihre Berichte in Ihre Applikationen integriert werden. Dies kann auf verschiedene Arten erfolgen, die im Folgenden erläutert werden. Zum einen können Sie den Bericht auf Knopfdruck erzeugen lassen, indem Sie die Erzeugung direkt an eine Schaltfläche koppeln, zum anderen können Sie den Bericht im Hintergrund automatisiert erzeugen lassen, bspw. für periodische Berichte, die per E-Mail verschickt werden sollen.

5.5.1 Schaltflächen-Aufruf: Klassische Variante

Die so genannte klassische Variante für den Schaltflächen-Aufruf bezieht sich auf die Vorgängerversionen von Reports für Intrexx. Sie beschreibt die Option, dass eine Berichtsdatei auf dem Server durch einen Sprung auf eine Seite erfolgt, in die das Velocity-Skript

 `<portal>\internal\system\vm\html\include\showreport.vm`

eingebunden ist.


Diese Funktion steht nicht mehr zur Verfügung. Da ab Intrexx 19.03 leider keine benutzerdefinierten Expert-Attribute an Schaltflächen mehr zugelassen werden und



außerdem die Sprungzielart *Popup* nicht mehr zur Verfügung steht, müssen Sie sämtliche klassischen Reports-Schaltflächen-Aufrufe auf das neuere Verfahren *5.5.2 JavaScript-Aufruf: Servlet* umstellen.

5.5.2 JavaScript-Aufruf: Servlet

Bei dieser Variante wird ein Report via JavaScript direkt vom Intrexx Server über ein eigenes Reports-Servlet angefordert. Der Vorteil der Servlet Variante ist, dass der Bericht direkt und ohne Zwischenspeicherung auf dem Server an den Benutzer als Binärdatenstrom ausgeliefert wird.

5.5.2.1 Import der JavaScript-Bibliothek

Falls noch nicht erfolgt, sollten Sie die im Installationspaket mitgelieferte QuinScape JavaScript Bibliothek importieren. Öffnen Sie dazu im Applikationsdesigner den internen Skripteditor (*F12*). Nach einem Rechtsklick auf den Knoten  *Bibliotheken* wählen Sie den Menüpunkt *Bibliothek importieren* aus. Wählen Sie die JavaScript-Bibliothek *qsreportsjslib.xml* vom Installationspaket, Ordner *Ressourcen*, *Skripte zum Importieren* aus. Anschließend steht Ihnen die QuinScape Reports-JavaScript Bibliothek zur Auswahl.

 **Hinweis:** Falls Sie bereits eine alte Version der Bibliothek importiert haben, so müssen Sie diese zunächst über einen Rechtsklick auf die entsprechende Bibliothek  *Löschen* entfernen, bevor Sie die neue Bibliothek importieren können.

5.5.2.2 Beispiel

Wechseln Sie in den JavaScript-Editor und fügen Sie das Template der JavaScript-Funktion `makeReportServletXhrCall(oHtmlActionControl, report)` zu Ihrer Applikation hinzu.

Definieren Sie anschließend eine Schaltfläche, die die o. g. Funktion aufruft.

Auf der Seite definieren Sie Eingabekontrollen mit folgenden Namen und Werten. (Den Namen setzen Sie über das Expert-Attribut *name*).

1. `qsExportFormat: <Format>`, in unserem Fall `PDF`.

Die JavaScript-Funktion

```
makeReportServletXhrCall(this, 'Benutzerverzeichnis');
```

muss beim Klick (*onclick*-Ereignis) aufgerufen werden.

5.5.3 Prozessmanager-Aufruf / *sysident*-Werte

Die Konfiguration des Berichts, seiner Parameter und der Dateifelder, in denen er gespeichert werden soll, über *sysident*-Werte wird **ab Reports für Intrexx 19.03 nicht**

mehr unterstützt. Bitte erweitern Sie ihre vorhandenen Groovy-Skript-Aktionen um diese Daten, die zuvor über die *sysident*-Werte automatisch ermittelt wurden.

5.5.4 Groovy-API Aufruf

Die Groovy-API ist eine flexible Möglichkeit zur Erzeugung eines Berichts mit Hilfe von Intrexx Prozessen. Der Bericht selbst und seine Parameter werden hier dynamisch per Groovy-Skript definiert. Im Folgenden werden Beispiele für den Aufruf eines Reports mittels der Groovy-API beschrieben.

- 1) Aufruf des Berichts *Telefonverzeichnis*, der in eine andere Datengruppe als die aufrufende unter Angabe des Zieldatenfelds sowie eines neuen Dateinamens gespeichert wird.
- 2) Aufruf des Berichts *Telefonverzeichnis*, der in *zwei Formaten* abgelegt wird und für den eine Reihe von *Parametern* programmatisch ermittelt werden.
- 3) Aufruf des Berichts *Telefonverzeichnis*, der in einem Format abgelegt wird. Das Besondere hierbei ist, dass dieser vorgelegte Werte (Elterndatensatz, Titel) sowie in einer Datengruppe mit zusammengesetztem Primärschlüssel gespeichert wird.

5.5.4.1 Beispiel 1

```

1 // Import der benötigten Klasse
2 import de.uplanet.lucy.server.spring.configuration.IntrexxApplicationContext;
3
4 // Instanziierung der benötigten Klasse für die Reporterstellung
5 def reportsBean =
6 g_springApplicationContext.getBean("quinscape.reports.datarecordReportCreator");
7
8 // Aufruf der Reporterstellung
9 reportsBean.createReport([
10
11     // Aktiviert das Debugging für diesen Aufruf in der Portal-Log-Datei
12     debug: true,
13
14     // Der Kontext für den Report (bedarf in der Regel keiner Änderung)
15     context: g_context,
16
17     // Der Identifier des aufzurufenden Reports (GUID oder Name)
18     identifier: "Telefonverzeichnis",
19
20     // Das Ziel des Reports, in diesem Fall ein Datensatz
21     datarecord: [
22
23         // Neuer Ziel-Datensatz
24         newRecord: true,
25
26         // Ziel-Datengruppe (GUID)
27         datagroupGuid: "FCD7F8634EC846A3223DF29B2D8E13172DA2FDB9",
28

```

```
29 // Ziel-Angaben für Datenfeld, Format usw.
30 datafields: [
31
32     // Definiert einen neuen Export
33     // Es wird ein PDF mit dem Namen „test<Feldinhalt>“
34     // erstellt und in dem Datenfeld FILE_DATENFELD_4B156821
35     // in der o.g. Datengruppe abgelegt.
36     [format: "pdf", filename: "test" +
37 g_record["B5477BFE8AA8A4DE8626A66617FC29475E793F0F"].getValue(), identifier:
38 "FILE_DATENFELD_4B156821"]
39
40 ]
41
42 ],
43
44 // Das Ziel des Reports, in diesem Fall eine Datei (aus dem Snippet
45 // Zielfeld). Dies hinterlegt den Report zusätzlich zu dem zuvor
46 // genannten Dateidatenfeld auf dem Dateisystem auf dem Laufwerk D:
47 // mit dem Dateinamen Test.xls im entsprechenden Format
48 files: [
49     [format: "xls", path: "D:/", filename: "Test.xls"]
50 ]
51
52 ];
```

5.5.4.2 Beispiel 2

```

1 // Import der benötigten Klasse
2 import de.uplanet.lucy.server.spring.configuration.IntrexxApplicationContext;
3
4 // Instanziierung der benötigten Klasse für die Reporterstellung
5 def reportsBean =
6 g_springApplicationContext.getBean("quinscape.reports.datarecordReportCreator");
7
8 // Definiert eine Variable, die später als Parameter genutzt wird.
9 def meineVariable = 3 * 4 * 5 + 1;
10
11 reportsBean.createReport([
12
13     // Aktiviert das Debugging für diesen Aufruf in der Portal-Log-Datei
14     debug: true,
15
16     // Der Kontext für den Report (bedarf in der Regel keiner Änderung)
17     context: g_context,
18
19     // Der Identifier des aufzurufenden Reports (GUID oder Name)
20     identifier: "Telefonverzeichnis",
21
22     // Das Ziel des Reports, in diesem Fall ein Datensatz
23     datarecord: [
24
25         // Ziel-Datensatz
26         recordId: 1,
27
28         // Ziel-Datengruppe
29         datagroupGuid: "FCD7F8634EC846A3223DF29B2D8E13172DA2FDB9",
30
31         // Ziel-Datenfelder
32         datafields: [
33
34             // Definiert einen neuen Export
35             // Es wird ein ODS mit dem Namen „test<Feldinhalt>“
36             // erstellt und in dem Datenfeld FILE_DATENFELD_001B5D85
37             // in der o.g. Datengruppe abgelegt.
38             [format: "ods", filename: "test" +
39 g_record["B5477BFE8AA8A4DE8626A66617FC29475E793F0F"].getValue(), identifier:
40 "FILE_DATENFELD_001B5D85"],
41
42             // Definiert ein weiteres Exportfeld
43             // Es wird ein PDF mit dem Namen „test<Feldinhalt>“
44             // erstellt und in dem Datenfeld FILE_DATENFELD_41242AS
45             // in der o.g. Datengruppe abgelegt.
46             [format: "pdf", filename: "test" +
47 g_record["B5477BFE8AA8A4DE8626A66617FC29475E793F0F"].getValue(), identifier:
48 "FILE_DATENFELD_41242AS"]
49         ]
50     ]
51 ],
52

```



```

53
54 // Definition von Parametern, die an den Report unter dem
55 // jeweiligen Schlüssel übergeben werden (abc, X, test, berechnet,
56 // login)
57 parameters: [
58
59     // Einfacher Wert
60     abc: 123,
61
62     // Datum: Jetzt minus 24 Stunden
63     X: new Date(System.currentTimeMillis() - 1000 * 60 * 60 * 24),
64
65     // Ein boolescher Wert
66     test: true,
67
68     // Vorher berechneter Wert
69     berechnet: meineVariable,
70
71     // Valueholder aus der Datengruppe
72     login: g_record["EBBF09B0C89616A648C67515E9116BA9B706F99B"]
73
74 ]
75 ];

```

5.5.4.3 Beispiel 3

```

1 // Import der benötigten Klasse
2 import de.uplanet.lucy.server.spring.configuration.IntrexxApplicationContext;
3
4 // Instanziierung der benötigten Klasse für die Reporterstellung
5 def reportsBean =
6 g_springApplicationContext.getBean("quinscape.reports.datarecordReportCreator");
7
8 // Aufruf der Reporterstellung
9 reportsBean.createReport([
10
11     // Der Kontext für den Report (bedarf in der Regel keiner Änderung)
12     context: g_context,
13
14     // Der Identifier des aufzurufenden Reports (GUID oder Name)
15     identifier: "Telefonverzeichnis",
16
17     // Das Ziel des Reports, in diesem Fall ein Datensatz
18     datarecord: [
19
20         // Neuer Ziel-Datensatz
21         newRecord: true,
22
23         // Angabe des zusammengesetzten Primärschlüssels des neuen Datensatzes
24         recordId: "LANG= DE, LID= 1",
25
26         // Weitere Werte des neuen Datensatzes
27         recordValues: [

```


```

28
29      // Der Elterndatensatz hat die ID 3
30      FKLID: 3,
31
32      // Das Feld "STR_TITEL" erhält den Wert "Mein neuer Report"
33      STR_TITEL: "Mein neuer Report"
34
35  ],
36
37      // Ziel-Datengruppe (GUID)
38      datagroupGuid: "FCD7F8634EC846A3223DF29B2D8E13172DA2FDB9",
39
40      // Ziel-Angaben für Datenfeld, Format usw.
41      datafields: [
42
43          // Definiert einen neuen Export
44          // Es wird ein PDF mit dem Standardnamen des Reports
45          // erstellt und in dem Datenfeld FILE_DATENFELD_4B156821
46          // in der o.g. Datengruppe abgelegt.
47          [format: "pdf", identifier: "FILE_DATENFELD_4B156821"]
48      ]
49  ]
50
51
52  })

```

5.5.5 Drill-Down

Mit der vorliegenden Version kann in Kombination mit dem Reports für Intrexx--Servlet ein echtes, direktes Drill-Down aus dem PDF heraus realisiert werden.

 Im Installationspaket und der Applikation Reportbeispieldaten finden Sie ein entsprechendes Beispiel, das wir im Folgenden erläutern. Das Beispiel bezieht sich auf Daten aus der Anwendung Reportbeispieldaten. Wenn diese Anwendung nicht in Ihr Portal importiert worden ist (sinnvollerweise mit Daten), wird der Beispielbericht nicht funktionieren.

Ein Drill-Down Bericht besteht aus mindestens zwei Einzelberichten, bei dem aus dem ersten Bericht auf Inhalte des zweiten Berichts verwiesen werden. In unserem Beispiel wird auf eine zweistufige Datenstruktur aus Firmen und deren Ansprechpartnern zugegriffen.

Der Bericht der Firmen-Daten (Hauptdaten) selektiert die Felder ID und Name des Hauptdatensatzes.

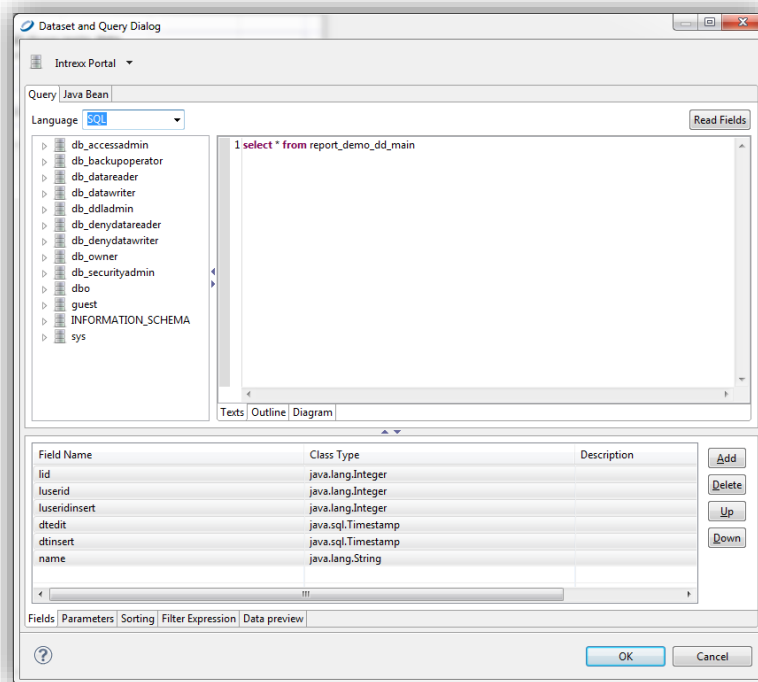


Abbildung 33 Abfrage des Hauptberichts

Drill-down main data			
Klick auf den Namen zeigt die zugehörigen Ansprechpartner (abhängige Daten).			
"ID"	"Name"	Column Header	
<code>\$F{lid}</code>	<code>\$F{name}</code>		
"Referring back to: " + <code>\$P{INTREXX_REQUEST}.SCHEME</code> + <code>"/"</code> + <code>\$P{INTREXX_REQUEST}.</code>			"Page "+ <code>\$V</code> " " + <code>\$V</code>

Abbildung 34 Platzierung der Felder im Hauptbericht

Das Feld `$F{name}` wird mit einem Hyperlink versehen, der das Reports-Servlet auf dem Server aufruft. Zu diesem Zweck können wir den impliziten Parameter `INTREXX_REQUEST` (siehe Kapitel 2.3.1.1.5) verwenden. Die aufzurufende URL:

```
$P{INTREXX_REQUEST}.REQUEST_URL + "?urn:schemas-unitedplanet-
de:ixservlet:name=reports&qsRe-
port=B74EFA534649ADAEF04F6512D28C0CBCEAA938D1&qsExportFor-
mat=PDF&fkId=" + $F{lid}
```

wird nach folgendem Schema aufgebaut:

`$P{INTREXX_REQUEST}.REQUEST_URL` wird verwendet, um die Server-URL für das Reports-Servlet dynamisch aus der Benutzeranfrage zu extrahieren.

`urn:schemas-unitedplanet-de:ixservlet:name=reports` ist der Name des Reports-Servlets

`qsReport=B74EFA534649ADAEF04F6512D28C0CBCEAA938D1` gibt den aufzurufenden Bericht (GUID) an

`qsExportFormat=PDF` definiert das Format, in dem der Bericht erzeugt werden soll

`fklid=" + $F{lid}` ist ein Parameter des Subreports. Hier wird die ID des Elterndatensatzes angegeben, für den wir die Kinddaten (abhängigen Daten) anzeigen lassen möchten. Dieser Parameter wird in der Datenbankabfrage des Zielberichts verarbeitet.

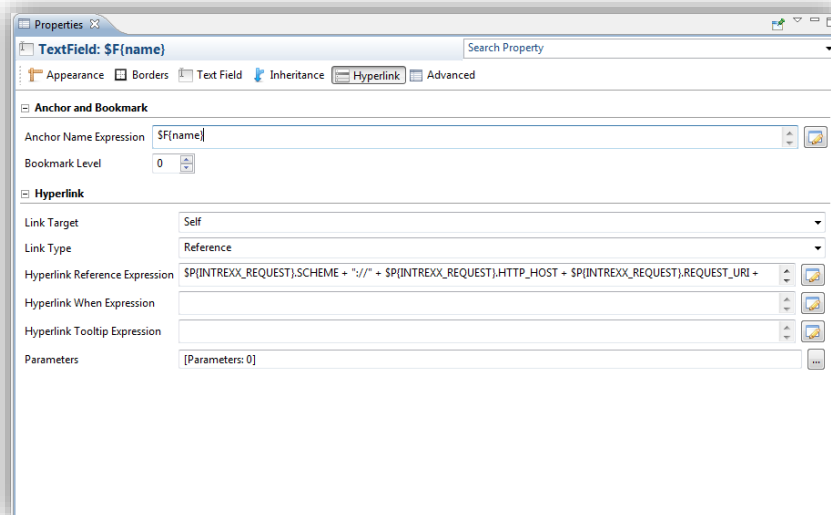



Abbildung 35 Definition des Hyperlinks


Der Hyperlink hat noch weitere Eigenschaften, die steuern, wie dieser sich öffnet. Das Ziel *Self* gibt an, dass das Ziel im aktuellen Dokument geöffnet werden soll. Alternativ könnte auch ein neues Fenster/ein neues Dokument geöffnet werden. Der *Hyperlink type Reference* ist zu wählen, um eine URL aufzurufen, wie in unserem Fall notwendig. Alternative (z.B. *local anchor*) können verwendet werden, um im aktuellen Dokumente zu einem Anker zu springen (selbst vergebene Sprungmarke). Ein Anker wird im oberen Bereich des Dialogs vergeben und definiert eine (eindeutige) Markierung im Dokument (bspw. ID des aktuellen Datensatzes).

Das Beispiel können Sie in der Anwendung *Reports für Intrexx – Beispieldaten* in Aktion sehen.

6 Weiterführende Informationen

6.1 Jaspersoft University sowie Coaching und Unterstützung bei der Berichtserstellung

QuinScape ist TIBCO Analytics Partner und bietet am Standort Dortmund die offizielle **Jaspersoft University** an. In unseren professionellen Schulungskursen erfahren Einsteiger wie auch Profianwender, wie sie ihre Produktivität in Jaspersoft-Anwendungen optimal steigern können. Die Präsenzkurse der Jaspersoft University bieten optimale Lernbedingungen, um sich mit sämtlichen Anwendungen und Funktionen der Jaspersoft Business Intelligence-Suite vertraut zu machen. Neben Vorlesungen umfassen diese Kurse auch praktische Übungen sowie typische Praxisfälle, in denen Teilnehmer ihr Wissen weiter vertiefen können. Unter  www.quinscape.de/jaspersoft-university finden Sie weitere Informationen sowie eine Übersicht der nächsten Schulungstermine in Dortmund.

Weiterführende Informationen, insbesondere zur Erstellung von Berichten erhalten Sie in der Schulung zum Thema Reports oder in Einzelseminaren. Über  www.quinscape.de/reports erhalten Sie die Kontaktinformationen für eine Einzelschulung. In der Regel wird Reports für Intrexx mit einer ein- oder zweitägigen Einweisung angeboten.

Darüber hinaus unterstützen wir Sie gerne auch bei der Berichtserstellung von der Konzeption bis zur Realisierung. Richten Sie einfach Ihre Anfrage an reports@quinscape.de, um ein Unterstützungsangebot zu erhalten.

6.2 Managed Services: Support und verlängerte Werkbank für Ihre Berichts-anwendungen und Berichte

Wird ein System in Produktion genommen, so muss ein sicherer Betrieb gewährleistet sein. Etwaige Störungen sind kurzfristig zu beheben – unabhängig von der Ursache. Auch wird erwartet, dass kleine Anpassungen und Wünsche kurzfristig im System umgesetzt werden, beispielweise müssen neue Berichte oder Dashboards kurzfristig erstellt oder weitere Datenquellen angebunden werden.

Die Herausforderung liegt auf der Hand: Wie kann die kurzfristige Verfügbarkeit von qualifizierten und mit dem individuellen System vertrautem Fachpersonal sichergestellt werden?

Unsere Supportleistungen für „Reports für Intrexx“- , Intrexx- und Jaspersoft-basierende Systeme halten Ihnen den Rücken frei. Wir unterstützen Sie vertraglich definiert auch kurzfristig mit qualifiziertem Fachpersonal, das sich direkt auf Ihr System aufschaltet, Störungen beseitigt oder Anpassungen vornimmt. Die Abrechnung erfolgt nach Zeit in sehr kleinen Einheiten (Viertelstunden). Sie können folglich bedarfsgerecht und schnell auf qualifizierte Entwickler zugreifen, ohne diese selbst vorhalten zu müssen.

Mit dem Supportvertrag bieten wir Ihnen einen definierten Zugang zu einer „verlängerten Werkbank“. Die Abwicklung erfolgt über ein Serviceportal. Dieses Portal gewährleistet strukturierte Prozesse und eine transparente, dauerhafte Dokumentation. Durch Budgetgrenzen haben Sie die Kostenkontrolle; eine quartalsweise Abrechnung und die teilautomatisierte Abwicklung über das Portal minimieren den organisatorischen Overhead.

Einen Supportvertrag bieten wir übrigens auch für Systeme an, die wir nicht selbst realisiert haben.


Ihr Nutzen

- Sicherheit im Betrieb durch kurzfristige Unterstützung
- Unterschiedliche Service-Level, aus denen Sie bedarfsgerecht wählen können
- Schnelle Reaktionen auf kleinere Anforderungen aus den Fachabteilungen
- Keine Notwendigkeit, eigene Ressourcen vorzuhalten (inkl. Vertretungsregelungen etc.)
- Dokumentation aller Vorfälle (inkl. Bearbeitungshistorie)
- Abrechnung gemäß Inanspruchnahme und Kostenkontrolle
- Minimierung des organisatorischen Overhead in der Zusammenarbeit mit uns als verlängerte Werkbank


Für Rückfragen stehen wir jederzeit gerne zur Verfügung.

Weitere Informationen finden Sie unter  www.quinscape.de/tibco-support.

6.3 Jaspersoft Studio und JasperReports Dokumentation

Der Hersteller des Jaspersoft Studios und JasperReports bietet eine (teils kostenpflichtige) Dokumentation an. Diese kann unter  <http://www.jaspersoft.com/ultimate-guides> bezogen werden.

6.4 Foren

Für Diskussionen stehen Ihnen unter  <https://www.unitedplanet.com/forum/> sowie unter  <http://community.jaspersoft.com/project/jaspersoft-studio> Foren zur Verfügung, in denen Sie sich mit anderen Reports-Anwendern austauschen können.

6.5 Angrenzende Lösungen

6.5.1 Jaspersoft Business-Intelligence-Suite


Seit über 10 Jahren setzt QuinScape auf Jaspersoft-Technologien, zunächst in der Rolle als OEM-Partner. Aus der Kooperation sind dieses erfolgreiche Produkt „Reports für Intrexx“ und eine Business-Intelligence-Lösung speziell für das Portal-CRM entstanden.

Mittlerweile ist QuinScape Systemintegrationspartner für Jaspersoft und setzt im Rahmen des Wachstumskurses strategisch auf diese Plattform. Unsere Kunden und Interessenten unterstützen wir bei der Einführung von BI Lösungen und deren Integration in eigene Produkte oder Projektlösungen und bei allen Fragestellungen rund um die Produkte von Jaspersoft.

Durch echte „Business Intelligence“ Funktionen, zum Beispiel in Form von Ad-hoc Auswertungen von Daten, können Sie Reports für Intrexx nach oben hin abrunden. Ziel ist, dass ein Anwender mit profunden Excel-Kenntnissen ohne Programmierung auch komplexe Datenstrukturen auswerten und hieraus Erkenntnisse ziehen kann.

Bei Reports für Intrexx liegt das Prinzip zu Grunde, dass versierte IT-Mitarbeiter Berichte erstellen, die nachfolgend von den Anwendern im Portal (oder über sonstige Prozesse, zum Beispiel den automatischen E-Mail Versand) genutzt werden. Dies ist für die Benutzer sehr einfach und eignet sich hervorragend für Routine-Auswertungen und Berichte mit hoher Komplexität und/oder ästhetischer Qualität. Ebenso lassen sich hiermit vielfältige andere Anwendungen umsetzen wie zum Beispiel die Generierung von Korrespondenz oder hochwertigen Druckansichten.

Die Jaspersoft Business-Intelligence-Suite hingegen verfolgt ein anderes Ziel: Durch eine Abstraktionsschicht werden die technischen Daten im System auf die Begriffswelt der Fachanwender abgebildet und potenziell mit Berechtigungen versehen. Die Fachanwender sollen dann in die Lage versetzt werden, eigenständig Auswertungen zu erstellen, um bedarfsgerecht die Daten zu analysieren. Dies kann zum Aufdecken von Zusammenhängen ebenso dienen wie zur Vorbereitung von Präsentationen und dem Untermauern von Vermutungen mit „harten“ Fakten.

JasperReports stellt nur einen Teil der Business-Intelligence-Suite von Jaspersoft dar. In ihrer Gesamtheit handelt es sich bei der Suite um eine umfassende Open-Source-Lösung für das Thema Business Intelligence. Mehr zu diesem Thema finden Sie auf unserer Website unter  <http://www.quinscape.de/jaspersoft>.

Auf Basis unserer umfangreichen Erfahrung mit Jaspersoft unterstützen wir Sie gerne bedarfsgerecht bei der Konzipierung, Umsetzung und Einführung sowie der stetigen Weiterentwicklung und dem Support Ihrer Business Intelligence Lösung oder bei der Einbettung in Ihre Anwendung. Bitte wenden Sie sich einfach formlos an reports@quinscape.de oder melden Sie sich dort gleich zu einem unserer regelmäßig stattfindenden Webinare an. Alternativ gewinnen Sie einen fundierten Eindruck von Jaspersoft in einem Kurs der


Jaspersoft University in Dortmund
( www.quinscape.de/jaspersoft-university).

6.5.2 Spotfire

TIBCO Software bietet ein integriertes Set an Funktionalitäten, das in der BI-Welt konkurrenzlos ist: TIBCO Jaspersoft für Embedded Reporting und TIBCO Spotfire für Data Discovery und Datenvisualisierung.


Mit Spotfire können unvergleichlich schnell Fragen analytisch beantwortet werden. Ob Sie ein Anfänger oder ein Analyseexperte sind – keine andere Lösung macht es Ihnen so einfach, schnell einzusteigen, Dashboards zu erstellen oder wichtige Erkenntnisse aus Analysen zu gewinnen. Recommendations bietet Ihnen intelligente Tipps und Vorschläge, z. B. für Best-Practice-Visualisierungen. So erstellen Sie Dashboards in Sekundenschnelle. Mit der nahtlosen Benutzeroberfläche können Analysten und Dateningenieur-e Analyseanwendungen für Business-Anwender erstellen – ohne Visualisierungen miteinander verknüpfen zu müssen und ohne zusätzliche Schritte.

Spotfire unterstützt sowohl kleine als auch große globale Implementierungen und bietet Automatisierungsfunktionen, eine zentrale Verwaltung sowie eine hohe Performance. Lernen Sie eine Selfservice-Analyselösung kennen, mit der Sie keine Abstriche bei der Datensicherheit und der Compliance machen müssen.

Mehr zu diesem Thema finden Sie auf unserer Website unter  <http://www.quinscape.de/spotfire> oder melden Sie sich dort gleich zu einem unserer regelmäßig stattfindenden Webinare an.

A Anhang

A.1 Übersicht der verwendeten JasperReports-Version für Reports für Intrexx

 **Hinweis:** Bitte beachten Sie, dass es für verschiedene Intrexx Versionen auch passende Reports für Intrexx Versionen existieren. Diese Reports für Intrexx Versionen sind an die jeweilige Intrexx Version gebunden und funktionieren nicht mit anderen Intrexx Versionen.

Intrexx	Reports für Intrexx	JasperReports
19.03	19.03	6.9.0
18.03	18.03	6.5.1
8.0	8.0.1	6.2.1
8.0	8.0.0	6.2.1
7.0	7.0.3	5.5.1
7.0	7.0.1	5.5.1
6.0	6.0.4	5.2.0
6.0	6.0.3	5.2.0
6.0	6.0.2	5.0.4
6.0	6.0.1	4.7.1
6.0	6.0.0	4.7.1
5.2	5.2.2	4.5.0
5.2	5.2.1	4.1.2
5.2	5.2.0	4.0.0
5.1	5.1.0	4.0.0
5.0	5.0.4	3.7.2
5.0	5.0.3	3.7.2
5.0	5.0.2	3.7.2
5.0	5.0.1	3.7.2
5.0	5.0.0	3.7.2